SPECTRUM

VILÁG 17. rész



S121

A/Anfractuos
Supernova
Super Hero
Advanced Pinball Sim.
The Quest for the...
...Golden Egg Cup
Martian Knock Out
Race Track
B/Crime Busters
Gunfighter
Hollywood Poker
Draconus
Matt Lucas
Labyrinth by Water
Skittles
Ghost Hunt

S122

A/The Realm
Riding the Rapids
2088
Dizzy Dice
Joe Blade 2
Olympimania
B/Final W. (Manager)
Braccio di Ferro
Winged Avenger
Type Rope
The Fury
Sword Slayer

S123

A/Summer Games...
...(8 events)
Hopping Mad
Capitan Sevilla...
...(part I-II)
B/Silent Shadow...
...(+part 5)
Death Dishonour
1943 (The Battle...
...of Midway)

S124

A/Pacman's Revenge
Mercenary 2
Vindicator (part 3)
River Rescue
Mr. Wimpy
B/Overlander
Boxing Manager
Wolfman I-II-III.
Copter
Depthcharge

S125

A/The Empire Strikes Back
Barbarian...
...(Mastertronic)
Microbot
Stunt Bike Simulator
Ski Star 2000
Draughts Master
B/A View to a Kill...
...(part 1-5)
Terror from the Deep
Security Shelter

S126

A/Gift from the Gods
Antics
Blockade Runner
Spellbound (Beyond)
Strangeloop
Psytraxx
B/The Lords of Midnight
3D Tank Duel
The Guardians
Wizard's Lair
Mrs. Mopp
Pi-Eyed
Cavelon

S127

A/Panamajoe
The Adventure of...
...St.Bernard
911 TS
Steve Davis' Snooker
Jonah Barrington's...
...Squash
Pud-Pud
Icicle Works
B/Bristles
Highway Encounter
The Fourth Protocol...
...(part 1-3)
One Man and His Droid
Daley Thompson's...
...Supertest I.

S128 (128K)

A/Glider Rider 128
Rasputin 128
Three Weeks in...
...Paradise 128
B/Knight Orc I. 128
Knight Orc II. 128
Fairlight I. 128
Fairlight II. 128

S129 (128K)

A/The Arc of Yesod 128
Yie Ar Kung Fu 128
Mutants 128
B/Nodes of Yesod 128
Spitfire 40 - 128
Moonstrike 128
Grange Hill 128

S130 (*)

A/Copier FM3
Copier FM 3.1
Super 50K Copy
80K Copy
File Copy
Super FF Copy
Omnicopy 2
Zotyocopy +
Zotyocopy #
Satancopy 4
Copy-Copy
TC-4
TC-5
TC-6
TC-7
Screen$ Copy
Special Copy
Zap Copy
Copy 1.
Copy 2.
Minitape 4
Minitape 5
B-Copy
Supercopy
Best Copy 2
Minicopy
Hypercopy
Multicopy
Cpy Copy
I&L Copy
Sinclair Copy
Sinclair LERM Copy
Speeder
Mastercopy
Multicopy 4
Maxim
TF Copy
Copy de Luxe
Copy V6
Keyfile V7
MR Copy
B/Turbo 1 Copy
Copy V11
Copy V12
Tape List
Slowloader
Satancopy 3
Maxim Turbo
R.Copy
Detail Copier
New Copy MSX
Ticy Copy
Super Raus Copy
Turbo Tape
'C' Copy
Conversor
A.E. Copy
New Save Load
New Save Load Code
SP.Masolo
I & L Break
Copier TM +
Multicopy 2
Ultrafile 4
Copy
Microdrive Copy
MD Cat
MD Sector
SPD (MD)
Coplima +80 Copy
Coplima Copy
Vidata Copy
Copy-Right
Chop-Up Copy

A 'SpV' 15. részében megjelent keresztrejtvény helyes megfejtései:
Vízsz.1.: BRAD BLASTS THE GALACTIC BARBARIANS, **Függ.**1.: BMX TRIALS, JUMP CHALLENGE, BMX RACERS
A 14. rész szerencsés nyertesei: B.P. - Miskolc (S72), CS.Á. - Budaörs (-), G.A. - Gyöngyös (S100), K.A. - Eger (S68), P.E. - Budapest XII. (S100); **nyereményüket postáztuk!**

FIGYELEM!

Egyéni program-kollekciók, válogatások összeállítását - a 7. részben közölt terveinkkel ellentétben - **továbbra sem tudjuk vállalni**, ezért megkérjük Önöket, kazettát ilyen célból **ne küldjenek** a címünkre! **Köszönjük megértésüket!**

1. Egy elfelejtett periféria

1

Mi az a *'WAFADRIVE'*? Mi az a *'WAFER'*? Ezeket a kérdéseket teszi fel nagyon sok hazai Spectrumos, főként a fiatalabbak, vagy azok, akik nem túl régen kerültek közeli kapcsolatba a Spectrum számítógéppel.
Ezeknek a kérdéseknek a felvetődése is azt támasztja alá, hogy egy nem ismert, ill. ha pontosabban akarunk fogalmazni: elfelejtett perifériáról van szó. Jelen bevezetőnk azt a célt szolgálja, hogy feltárjuk az okokat: valóban megbukott a *'WAFADRIVE'*, s ha igen, akkor mi vezetett erre a sorsra?

A *Rotronics Ltd.* és a *Santosh House* közös fejlesztése 1984-ben látott napvilágot, mint a Microdrive konkurrenciája. A ***'WAFADRIVE'* egy** - a Microdrive-nál többször nagyobb méretű - nagy átviteli sebességű **háttértároló eszköz**, a tárolást egy - a microdrive cartridge-hez - hasonló felépítésű, speciális mágnesszalag kazettán (*'WAFER'*) oldották meg, a meghajtó: *DUAL BSR "stringy floppy"*. A *'WAFADRIVE'* **háttértároló sajátosságán túl Interface szerepet is betölt** (nincs szükség külön Interface-re, a géphez való csatlakoztatás esetén), mindemellett **Centronics** ill. **RS-232** kompatibilis nyomtatók is meghajthatók a segítségével. Bővített BASIC Interpretert tartalmaz, amely hatékony file-kezelést tesz lehetővé.
A *'WAFADRIVE'*-hoz a cég ingyen forgalmaz(ott) két cartridge-t, az egyik háttértárolón elhelyeztek egy hatékony szövegszerkesztő programot is.

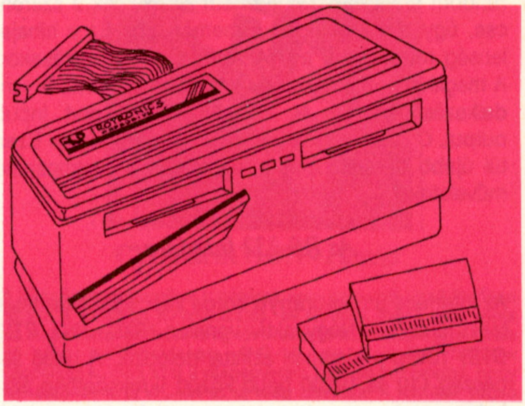
Az operációs rendszer

A *'WAFADRIVE'* operációs rendszere a '**WOS**', amely a DOS-hoz hasonló. Az operációs rendszer 8K ROM területe az alap ROM alsó 8K szeletére lapozódik be, a rendszer 26 új hibaüzenetet tartalmaz. Az inicializálás a NEW* utasítással történik. Az inicializálást követően a Spectrum memóriájából 2K RAM foglalt lesz a *'WAFADRIVE'* saját rendszerváltozói (23734-23825), írás/olvasás puffer (23836-24861) és a directory-k számára. Ez utóbbi azért lényeges, mert a *'WAFADRIVE'* perifériába egyidőben két cartridge-t helyezhetünk, azaz a két meghajtóról két külön directory-t (katalógus) kérhetünk (A: 24862-25443, B: 25444-26025). A '**WOS**' parancsai a hagyományos Spectrum BASIC parancsok '*' ill. '#' karakterrel történő kiégészítéseből képződnek.

A 'WAFER' és alkalmazása

A cég **3 féle cartridge**-t hozott forgalomba: 16K, 64K és 128K kapacitással. A file-ok elérési ideje fordítottan arányos a kapacitás nagyságával. A 16K kapacitású *'WAFER'*-eket célszerű programfejlesztések idején felhasználni, rövidebb rutinok tárolásához, míg a nagyobb kapacitású '**WAFER**'-ekre fésülhetjük össze a kész programot, ill. ezeken rögzíthetünk hosszabb játék és felhasználói programokat is. A szalag - a microdrive cartridge-ekkel ellentétben - a video kazettákhoz hasonlóan **védett**, csak akkor nyílik ki az író/olvasó nyílás, ha a cartridge-t a *'WAFADRIVE'*-ba helyeztük. Az aktuális *'WAFER'* működését LED is jelzi. A *'WAFER'*-t csak akkor szabad eltávolítani a *'WAFADRIVE'*-ból, ha a meghajtó motor már megállt. Ez hardware hibából gyakran nem következik be, ilyenkor a 'BREAK' billentyű használata hatásos lehet. A cartridge-ek kijelölése a DOS-hoz hasonló módon az 'a:' ill 'b:' szintaktikával történik, minden cartridge művelet esetén. Pl. az 'a:' egység formázása 'SpV' névre, a következőképpen történhet:

FORMAT *"a:SpV"



Formázáskor kialakulnak a szalagon a szektorok, majd a rendszer ellenőrzi a formázás hibátlanságát, végül kiírja a katalógust. Az elérési idő 16K-s 'WAFER' esetén 6.5 másodperc, míg 128K-s 'WAFER' esetén elérheti a 45 másodpercet is. A beolvasás sebessége másodpercenként 2 kbyte (18000 baud). A katalógus a 24862 címtől tárolódik el a memóriában. A '**CAT #**' utasítás esetén az aktuális meghajtó csak a memóriába olvassa be a katalógust, míg a '**CAT ***' utasítás kiadása esetén az a képernyőn is megjelenik. A directory - a microdrive-tól eltérően - részletes katalógus, a drive és *'WAFER'* nevén túl listát kapunk a file-ok (max. 32 - 16K-s *'WAFER'* esetén max. 16) neveiről, a file-ok típusáról (program, adat vagy gépi kód), a file-ok méretéről (kbyte-ban), a file-ok sorszámáról, a *'WAFER'* kapacitásáról és végül a még rendelkezésre álló szabad memóriahely nagyságáról (kbyte-ban). A file nevek rövidítve is megadhatók, ill. string-lánc '*'-gal helyettesíthető. A két meghajtó között adatátvitelre is lehetőség van a

'MOVE *"a:filenév" TO "b:"

szerkezet segítségével.

Stream-ek kezelése

Az alap rendszer csatornáihoz egy-egy stream lett rendelve (**K** - keyboard input/output - stream #0/1, **S** - screen output - stream #2, **P** - Sinclair nyomtató output - stream #3). A 16 rendelkezésre álló, és megnyitható stream közül a 4-15 intervallumba eső stream-ek definiálása esetén gyakran problémába ütközhettünk. Az alap Spectrum ROM rendelkezik egy súlyos hibával: egy meg nem nyitottt stream lezárása

esetén több dolog történhet, szerencsés esetben random hibát kapunk, ha nem áll mellénk a szerencse, úgy elszáll minden a memóriából. Ez a hibajelenség **megszűnik** a *'WAFADRIVE'* felhasználásakor. A meglévő csatornákon túl két újabb csatorna is rendelkezésünkre áll: **R** - RS-232 soros port input/output, **C** - Centronics párhuzamos port output. Ezen túl újabb parancsok is támogatják az adatmozgató műveleteket.

Az RS-232 Interface

Az interface un. **kétirányú soros** kapcsolat létesítését teszi lehetővé. Négy aktív vonalán (*RXD, TXD, RTS, CTS*) 10V p-p kimeneti szinttel dolgozik. A baud sebesség 110 és 19200 baud között 9 fokozatban állítható.

A Centronics Interface

Egyirányú párhuzamos kapcsolat létesítését teszi lehetővé. 26 pontos csatlakozóján egy byte adatait egy időben 8 vonalon jeleníti meg. *Centronics standard* bekötést valósítottak meg, azaz a 26 pontos illesztés 11 aktív vonalat tartalmaz (*Ground, Data Strobe, Busy, Data 0-7*).

Végső kiárusítás?

Napjaink angol számítástechnikai sajtótermékeit lapozgatva megakadhat a szemünk egy szenzációs hirdetésen: "**ROTRONICS *WAFADRIVE* CSAK 14.99 Font**". Nos, amikor egy hagyományos Centronics illesztő a mai napig 30-40 Fontba kerül még Angliában is, akkor igencsak elgondolkoztató az előzőekben ismertetett perifériáért ez az ár (15 Font kb. 1500 Forint). Megérzésünk szerint a periféria már megjelenése idején - a hajlékony mágneslemezek világában - sem volt modernnek mondható. Egy-egy cartridge még a microdrive cartridge-nél is drágább, sőt az utóbbi időben még a szigetországban is egyre nehezebben beszerezhető. A file elérési idő igen nagy értékű, a 8K ROM több jelentős hibával rendelkezik, a BASIC terület eltolódása a játékok 7o %-nak futtathatóságát lehetetlenné teszi, nem beszélve arról, hogy nem kompatibilis a 128K-s Spectrumok egyik verziójával sem. A periféria esztétikailag sem felel meg a kívánalmaknak, hordozhatósága is nehézkesebbé válik, hiszen tömege kb. 1 kg. Végkonklúzióként megállapíthatjuk, hogy az olcsó ár nem véletlen, nem célunk a *ROTRONICS Ltd*. termékét ócsárolni, ám **óva intünk mindenkit a periféria esetleges beszerzésétől.**

Nigel Mansell's GP

A játékban néhány dolog módosítására lehetőségünk nyílik, ha a betöltőt átugorjuk, s egy újat írunk helyette:

```
10 BORDER 0: INK 0: PAPER 0: CLEAR 24999
20 LOAD ""SCREEN$: POKE 23739,111: LOAD ""CODE: LOAD ""SCREEN$
30 FOR n=33580 TO 33595: POKE n,x: NEXT n
40 FOR n=33612 TO 33627: POKE n,y: NEXT n
50 DIM a$(20)
60 FOR n=1 TO 8: READ a$
70 FOR m=1 TO 20: POKE 53513+24*n+m,CODE a$(m)
80 NEXT m: NEXT n
90 RANDOMIZE USR 51795
101 DATA "Stefan Johansson"
102 DATA "Nigel Mansell"
103 DATA "Nelson Piquet"
104 DATA "Ayrton Senna"
105 DATA "Gerhard Berger"
106 DATA "Alessandro Nannini"
107 DATA "Thierry Boutsen"
108 DATA "Alain Prost"
```



A 30. sorban az 'x' helyébe írt szám lesz az összes pályán a max. 'QUALIFING TIME'. Az persze ne zavarjon senkit, hogy a gép az eredeti időket fogja kijelezni a képernyőn! A 40. sorban az 'y' tartalmazza az 5 km-re számított üzemanyag mennyiségét. A 101-108. sorokban átírhatjuk a játékosok neveit (egyenként max. 20 karakter hosszon).

Pyramid

Az örökélet beviteléhez töltsük be a BASIC loader-t, majd BREAK, és állítsuk meg a magnetofont.
Irjuk be: 285 POKE 44678,201 (ENTER)
Futtassuk ismét a betöltőt: RUN (ENTER) és indítsuk tovább a magnetofont.



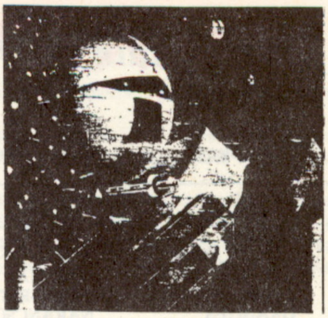
ANFRACTUOS - *Players*

A **Players** az utóbbi időben több színvonalas játékprogrammal lepett meg bennünket. Az űr mélyén fekszik Anfractuos - egy vastag kérgű planetoid, ahol valaha értékes ércet (Cryanium-ot) bányásztak. A naprendszer instabillá vált, s a bolygón lakókat félelem töltötte el, mi lesz, ha a nap Supernova lesz, vagyis kitágul, hiszen ez a bolygó pusztulását vonja maga után. Az embereket evakuálták, elhagyták a bányákat is, ám eközben egy elvetemült bányász droid megpróbált megszökni. Mi személyesítjük meg a szökésben lévő droid-ot. Ahhoz, hogy eltudjunk szökni, be kell gyűjtenünk 8 energia egységet, de ez nem könnyű. Az őrökkel való minden találkozás fogyasztja az ezergiánkat. Az őrök védettek, ezért legyünk óvatosak, amikor lőjük őket, vagy később megfizetünk érte. Az energia-csomagok az energia töltő állomásokon tölthetők fel, a bánya környékén. Használhatjuk menet közben a lifteket is, de a karbantartás hiánya miatt a berendezés gyakran meghibásodhat. Ahhoz, hogy a rakétával távozni tudjunk a bolygóról, 4 kapcsolót kell megtalálni, majd aktivizálni. Ekkor beindul egy visszaszámlálás. Sietnünk kell az űrhajóhoz, mert ha lassúak vagyunk, az nélkülünk távozik. A játék, amelyből egyébként kiléphetünk a 'J+K+L' billentyűk együttes lenyomásával, izgalmas időtöltést ígér mindenki számára.



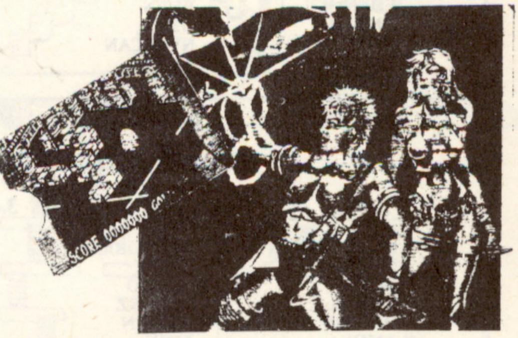
SUPERNOVA - *Players*

Egy újabb **Players** játék, amely ötvözi a Sci-fi-, az akció-, és a szerencsejátékok elemeit. Az automatizálás világában még az intelligens robotoknak is szükségük van arra, hogy kikapcsolódjanak az egész éves 'robot'-ból. A robotok 'felső tízezre' szívesen tölti éves szabadságát az Antares bolygó Supernova hoteljében, ahol olyan luxuskörülmények között pihenhetnek, amihez hozzászoktak. Annak ellenére, hogy az itt töltött szabadság sokkal többe kerül, mint a jövedelmük, robotunk mégis mindenféleképpen itt akarja tölteni a szabadságát. A robot megrögzött optimista, az árak mégis magasabbak, mint ahogy azt remélte. A szabadság végére főhősünknek jóformán semmilye sincs, csak 'napbarnított bőre' és egy hatalmas tartozás a számláján. A hotel személyzete egyre idegesebb lesz, és úgy dönt, hogy a robot nem mehet el addig, amíg ki nem fizette a számláját. Habár a hotel egy 'menyország' az űrben, a robot honvágya mégis fokozódik, tehát segítenünk kell rajta, hogy összegyűjtse a tartozását és hazatérjen. A hotelben lakó túristák könnyen rászedhetők, így a menekülés egyetlen módja a rablás, hamisítás, űzérkedés ill. a tiltott szerencsejáték lehet...
Érdekes stratégiát igényel a megoldás, ennélfogva hosszan tartó jó szórakozást ígér.



CYBERNATION - *Players*

A különösen fantasztikus történetek sora ismét bővült. A jövőben járunk, a rohamos automatizálás, és fejlődés maga után vonja az emberi szükségletek megnövekedését is, ezáltal egyre újabb és több mellékhelyiségre (WC-re) van szükség. Tudósok csoportjait küldték szét a világegyetem kísérleti intézeteibe, hogy fejlesszenek ki új WC-ket az új világ számára. A tudósok végső konklúziója az volt, hogy egy szervezetre van szükség, amely az egész galaktikát felszereli modern lefolyókkal és öblítőkkel ellátott népi klotyókkal. Ezen előzmények után végül is megalakult a **GUTA** (Galactic Union of Toilet Attendants = WC Kiszolgálók Galaktikus Egyesülete). A CYBERNATION állomáson az egyesület hűséges dolgozói rosszul funkcionálnak, nem tudnak felébredni szokásos téli álmukból, ez pedig gondot jelent, mert ha a nyári szezonra nem lesznek üzemkészek a WC-k, az beláthatatlan következménnyel járhat.
Mi vagyunk a **GUTA** utolsó reménysége, kapunk egy SQUAM teleport-ot, és feladatunk, hogy az állomáson felébresszük az alvó személyzetet. A WC-k a bolygókon körbe-körbe lettek elhelyezve. A kiszolgáló személyzet egy tagjának felkeltését követően meg kell oldanunk egy betű-rejtvény feladatot is.
Különös fantázia és logika együttese ez a játék, valószínűleg hamar megkedveljük mindannyian.



SUPER HERO - *Code Masters*

Akik ismerik a **BAT MAN** ill. a **HEAD OVER HEELS** grafikáját és stílusát, bizonyára nem fognak csalódni ebben a játékban sem, hiszen grafikáját az előbb felsorolt játékok grafikusa, **BERNIE DRUMMOND** tervezte. Hagyományos '3D szobás' játék, újabb elemekkel. Legelőször is célszerű felderíteni a terepet, majd meg kell szereznünk a négy varázslatot, mert csak így rendelkezhetünk természet feletti képességekkel. Az első varázslat 'POWER BOOTS' (energia csizmák) az ugrásainkat fogja segíteni, a második 'RETURNING WARHAMMER' segítségével megölhetjük a szörnyeket és a vámpírokat a folyosókon, a harmadikkal 'TRANS-DIMENSIONAL BAG' (többdimenziós-csomag) segítségével megsemmisíthetünk egy tárgyat, míg végül a negyedik varázslat 'MYSTICAL HELMET' (misztikus sisak) az 'isteni ajándékokat' teszi láthatóvá. Ezek közé tartoznak a 'TELEPORT KEYS' (teleport kulcsok), amely a teleportálást segíti elő, a 'MERCURY'S WINGED SHOES' (Merkúr szárnyas cipői) segítségével gyorsabban futhatunk, a 'PARALYSER' eltünteti a szörnyeket a folyosókról, a 'GOLD' (arany) segítségével vásárolhatunk a kereskedőknél, az 'ESP ACTIVATORS' segítségével eljuthatunk a nagy szellemhez (SPIRIT), a zöld jelzés a jó, a vörös jelzés a rossz irányt mutatja. Reméljük, hogy a játék kiváló grafikája, és az izgalmas kaland nemcsak a **BATMAN** rajongók népes táborát fogja magával vonzani!

STARION 3/1-4 zónák

3.1 zóna

Honnan	Angol jelentés	Magyar jelentés	Hova	Probléma megnevezése
1.	OIL	OLAJ	4.	Kezdetl. felfedezés Pennsylvániában
4.	STANLEY	STANLEY	7.	"Dr Livingstone, I presume?"
7.	COMET	ÜSTÖKÖS	8.	Sugárhajtású repülőgép csóvája
8.	CASABLANCA	CASABLANCA	2.	"Játszd újra Sam"
2.	HELENS	HELÉNA	5.	A szent hegy kitör
5.	TERRORIST	TERRORISTA	9.	Müncheni zsákmány-sport
9.	ORANGES	NARANCSOK	6.	Prokofjev hármat szeret
6.	HOOVER	PORSZÍVÓ	3.	Az átkozott jó elnök
3.	PSYCHO	LÉLEK	1.	Hitchoch őrült ember

A zóna azonosítója: *HOPSCOTCH*

3.2 zóna

1.	OBERAMMERGAU	OBERAMMERGAU	7.	A szenvedélyes játék itt kezdődik
7.	NYLON	NYLON	6.	NY-i és Londoni szintetikus anyagok
6.	WHEEL	KERÉK	9.	Sarkalatos találmány
9.	EQUATOR	EGYENLÍTŐ	4.	Ausztrália felé átlépi ezt a vonalat
4.	TRANSISTOR	TRANZISZTOR	3.	Walter Britain apró erősítője
3.	AMBER	BOROSTYÁNKŐ	8.	A hiányzó jezőlámpa
8.	VISA	VÍZUM	5.	M. Polo belépési engedély kérelme
5.	FAWKES	FAWKES	2.	Fickó puskaporral
2.	RAGTIME	RAGTIME	1.	Zavaros beat a jazz-ben

A zóna azonosítója: ?????????

(A blokk azonosítás a mi verziónkban itt elmaradt, továbbléphettünk!)

3.3 zóna

1.	SINGER	SINGER	8.	Choral első varrógépe
8.	TELEVISION	TELEVÍZIÓ	4.	Nézd a BBC-t
4.	EDEN	ÉDEN	6.	Ádám és Éva a kertben
6.	DNA	DNA	5.	Deoxyribonucleid acid
5.	EIGHTS	NYOLCAN	2.	Halálosan forró egyetemista evezősök
2.	ALICE	ALICE	7.	Kalandjai csodaországban
7.	RINK	JÉGPÁLYA	3.	T és DS a Lake Placid-i szinpadon
3.	YORKTOWN	YORK VÁROSA	9.	A britek megadása a függetlenségi háborúban
9.	YEAGER	JÉGER	1.	Chuck szuperszónikus az elbűvölő völgyben

A zóna azonosítója: *YESTERDAY*

3.4 zóna

1.	MULL	BOR FORRALÁS	7.	Kintire szárnyaló bora
7.	KNIVES	KÉSEK	3.	Hitler hosszú fegyvere az éjszakában
3.	ROMEO	RÓMEÓ	8.	Ahol te vagy a művészet
8.	REMUS	RÉMUSZ	9.	Romulusz párja
9.	OXIGEN	OXIGÉN	5.	Túl sok van Apolló tüzében
5.	AMIN	AMIN	6.	Ugandai tömeggyilkos
6.	FISH	HAL	4.	Ötöt ötezerért
4.	ELGAR	ELGAR	2.	Változata egy rejtély
2.	WING	SZÁRNY	1.	Bleriot csatornákat nyit maga alatt

A zóna azonosítója: *FRAMEWORK*

Cybernoid

Az örökélethez töltsük be a BASIC loader-t, majd a SCREEN-t, sőt még a főprogram fejlécét is, és itt állítsuk meg a magnetofont, párhuzamosan RESET-eljük a gépet is. Írjuk be a következő BASIC programot:

```
5 CLEAR 25000
10 FOR f = 23296 TO 23309: READ a: POKE f,a: NEXT f
20 DATA 221,33,4,101,17,252,154,62,255,55,205,86,5,201
30 RANDOMIZE USR 23296
40 POKE 27330,0
50 POKE 29215,0: POKE 29216,0: POKE 29217,0
60 RANDOMIZE USR 25860
```

Adjuk ki: RUN (ENTER), majd indítsuk a magnetofont.

BEYOND THE ICE PALACE (Elite)

A Spectrum Világ 11.részében már bemutattuk ezt a színvonalas programot. Most ismét közlünk a játékhoz néhány kulcsfontosságú információt.

Az út során különböző erejű fegyverek találhatók. A kritikus helyzetekben megidézhető az erdő szelleme, amely megöli, illetve legyengíti a támadókat. Csínján kell vele bánni, mert kezdetben csak két ilyen lehetőségünk van, bár a játék folyamán szerezhetünk még. Egyszerre azonban mindig csak legfeljebb kettővel rendelkezhetünk.

A játék irányítható joystick-kel, illetve a billentyűzet is átdefiniálható. A BREAK a játék végét jelenti. (Summon Spirit = szellemidézés)

A kard a legerősebb fegyver. A buzogány használható a felső platformokon lévő ellenségekkel szemben. Ha egy gyengébb fegyvert (tőrt) vettünk fel, próbáljunk visszamenni kardért. Viselkedjünk férfi módjára: vegyük fel a harcot. A menekülés nem segít, az ellenségek követnek bárhová. Ne rohanjunk vaktában előre, mert hamar bekerítenek. Ha felmászunk a létrán, nagyon ügyeljünk a fentről támadókra. A liften figyeljünk a denevérekre: egy gyors szellemidézés segíthet.

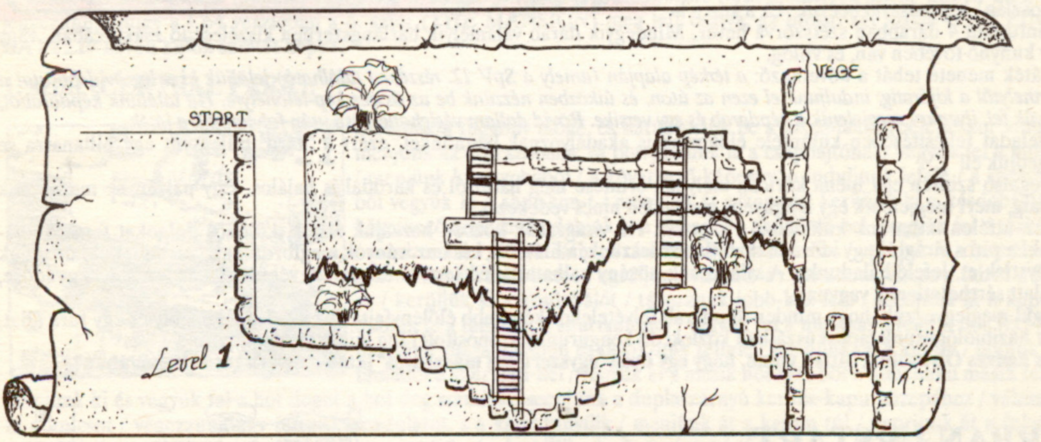
A játék megoldását néhány POKE elősegítheti:

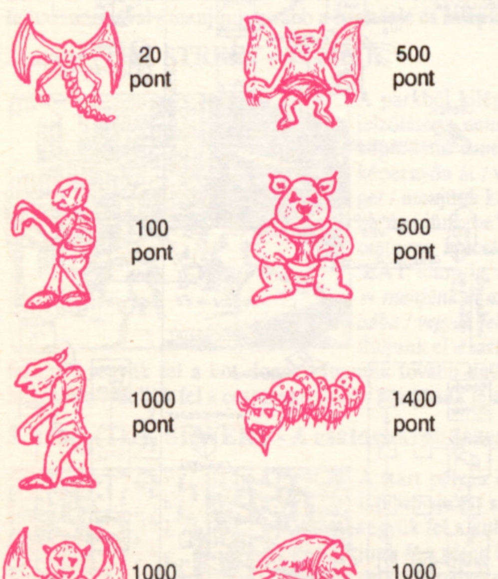
POKE 38279,0: POKE 38280,0: POKE 38281,0 - **örökélet**

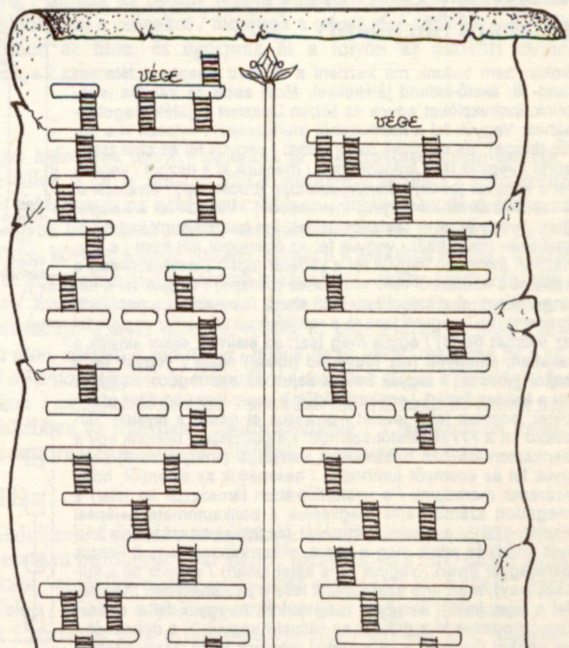
POKE 37809,0: POKE 37810,0: POKE 37811,0 - **végtelen sok szellem**

A POKE-ok a játék betöltőjébe egyszerűen beírhatók. A SCREEN$ és a CODE töltését követően írjuk be a POKE-okat, még a RANDOMIZE USR 24832 utasítás előtt.

Beyond the Ice Palace térkép:







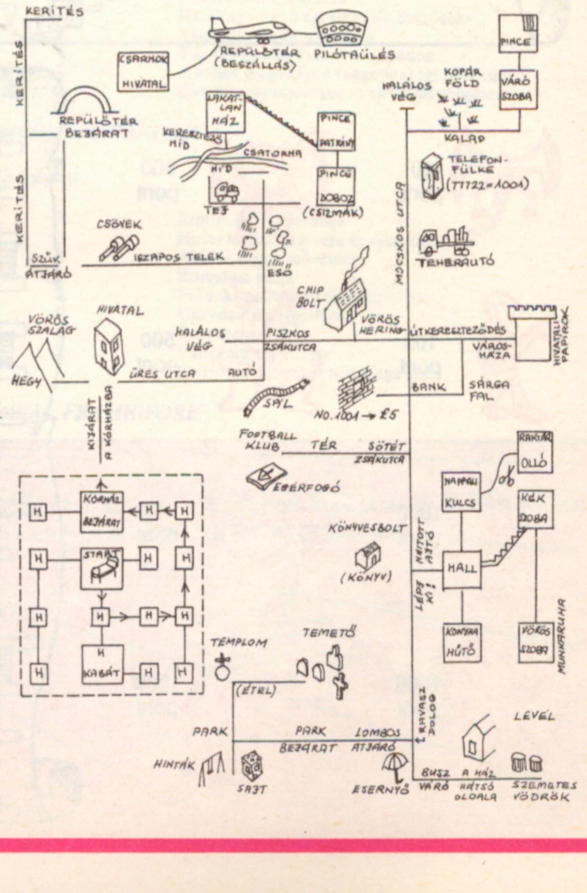
SABRE WULF (Ultimate/A.C.G.)



1985-ben jelentet meg az **ULTIMATE - A.C.G.**-csoport *Sabre Wulf* című játéka, amely azóta klasszikussá vált. Egy akkoriban kibontakozó stílus talán legsikeresebb tagja volt: szép, ötletes grafika, ragyogó háttérszínek, több száz képernyő, és izgalmas órák a játékos számára.
Ebben a játékban egy **256 szobás dzsungel-labirintusból kell kijutnunk**. A startszobából kilépve rögtön meglátjuk a kijáratot - egy fal túlsó oldalán. Nem léphetünk be, ugyanis ha ezt megtennénk, igen kellemetlen meglepetés érne bennünket: az ajtóban áll egy furcsa figura, akinek az a rögeszméje, hogy csak az mehet tovább, akinél - egészen véletlenül - van egy festmény. Rembrandt alkotások híján kénytelenek vagyunk egy sörényes állat (oroszlán) képmásával beérni. Ez a kép a labirintusban 4 darabban szétszórva hever. Mindegyik darab valamelyik barlangszerűen kiszélesedő zsákutcában, vagy egy kunyhó tövében van, és villog.
A játék menete tehát a következő: *a térkép alapján (amely a SpV 12. részében található) jelöljük ki a legrövidebb utat a starthelytől a kijáratig; induljunk el ezen az úton, és útközben nézzünk be az összes kép-lelőhelyre. Ha találunk képdarabot, vegyük fel, ilyenkor megjelenik a képdarab és egy versike. Rövid dallam végighallgatása után folytatódik a játék.*
A feladat teljesítésében különféle élőlények is akadályoznak bennünket, ezért a "kard" billentyűt egy pillanatra se engedjük el!
A legalsó szinten egy hiéna kóricál, melynek érintése még hátulról és karddal is halálos. Egy pályán ne maradjunk sokáig, mert megjelenik egy lángnyelv, mely ellen nincs védekezés.
Úton-útfélen láthatunk különböző virágokat. Ha virágzásuk közben hozzájuk érünk, különös dolgokat tapasztalhatunk: a piros virágtól egy időre halhatatlanok leszünk, a lilától is, csak az irányok megfordulnak (pl. ha felfelé nyomjuk a joystick-et, lefelé haladunk). A világoskék növény halhatatlanná tesz és felgyorsít, végül a fehér "elkábít" egy időre, ezalatt sérthetetlenek vagyunk.
Végül megjegyezzük, hogy minden képdarab felvételekor két újabb élőlényfajta (pl. az elsőnél bagoly és egy fura állat, akit házibiológusunk laposkúszásban közlekedő kengurunak azonosított) fog minket boldogítani.
Nos kedves Olvasóink, hitték volna, hogy egy ilyen "egyszerű kis mászkálós" játék ennyi titkot rejt magában?

URBAN UPSTART
(Richard Shepherd)

Sokan nem tudunk mit kezdeni a Richard Shepherd féle kaland- ill. akció-kaland játékokkal. Most ebbe az irányba is lépünk, instrukciókat adunk az Urban Upastart c. játék megoldásához: Vegyük fel a munkaruhát (dungarees) / nyissuk ki a hűtőt (fridge), de ne igyuk meg a sört! / vegyük fel az ollót (scissors) / vegyük fel a kulcsot (key) / menjünk ki a házból / vegyük fel a könyvet (book) a könyvesboltban (bookshop) / olvassuk el a repülési útmutatót (flying instructions) / vegyük fel a kalapot (hat) / vegyük fel a szalagot (tape), ezt használnunk kell a városházán (town hall) / vegyük fel az egérfogót (rat trap) / a kórházban (hospital) vegyük fel a kabátot (coat) / viseljük (wear) a kabátot a kórházból való szökéshez (escape) / vegyük fel a heringet (herring) a halboltban (fish shop) / vegyük fel a papírokat (papers), szükségünk lesz rá a repülőtéren (airport) / vegyük fel az élelmet (food) / együk meg (eat) az élelmet, ekkor kinyílik a lakatlan, elhagyott ház (deserted house) ajtaja / vegyük fel a sajtot (cheese) / tegyük bele a sajtot az egérfogóba / vegyük fel a levelet (letter) / amikor elérjük a szemetes vödröket (dustbins), bontsuk fel a levelet / olvassuk el (read) a levelet: "tárcsázd fel a 77722 telefonszámot!" / a borítékban találunk egy a bankautomatákban használatos kártyát is (cheque card) / vegyük fel az esernyőt (umbrella) / használjuk az esernyőt, hogy szárazak maradjunk / a telefonfülkében tárcsázzuk fel (dial) a megadott számot, erre megkapjuk a bankautomata belépési kódját: 1001 / a bankautomatánál (cashline) használjuk a kártyát (card) és adjuk meg a kódot, ekkor kapunnk egy 5 Fontos bankjegyet (fiver) / vegyük fel a sálat (scarf) / vegyük fel a kulcsot (key), mert erre szükségünk lesz a pilótafülkében / vegyük fel a tejet (milk), és igyuk meg (drink) / vegyük fel a dobozt (box) / nyissuk ki a dobozt az ollóval / vegyük ki a dobozból a csizmákat (boots) / a csizmákkal menjünk át az iszapos telken (muddy site) / vegyük fel az öltönyt (suit), amelyet megtalálhatunk a csövek (pipes) segítségével (?!) / .... / várjunk, amíg csörög (ring) a telefon / adjuk oda a papírokat és az 5 Fontos bankjegyet a hivatalnoknak az indulási csarnokban (departure hall) / a pilótafülkében (cockpit) helyezzük be a kulcsot az indító nyílásba / 100 %!!!

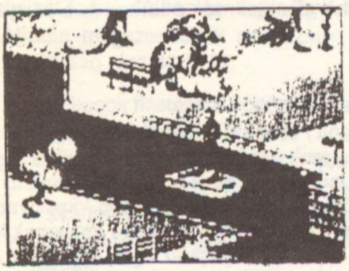


2.1 THE LAST NINJA 2.



Az utóbbi évek egyik legnagyobb sikerét aratta ez a program, különösen a Spectrum hívők táborában, hiszen a Spectrum számítógépre **nem készült el csak a történet második része**, s ennek megjelenése is több hónapot váratott magára. A fogadtatás lenyűgöző volt a szigetországban és hazánkban egyaránt, mind a Spectrum, mind a C64 tulajdonosok számára hosszan tartó fejtörést okozott a játék egyes részleteinek megoldása, megfejtése. Talán ennek köszönheti népszerűségét, különös tekintettel arra, hogy a hagyományos akciójátékokkal ellentétben itt gondolkodni kell, az egyes szintek teljesítése nem könnyű. Elnézést kérünk mindazoktól, akik haszontalannak tartották az eddigi ötleteket és térképeket, őszintén megvalljuk, számunkra sem volt könnyű a játék minden részletre kiterjedő feltárása, most pótolni szeretnénk az elmaradtakat. Mivel kb. egyforma arányban kaptunk leveleket Spectrum és C64 tulajdonosoktól, melyekben tanácsokat kérnek tőlünk, ezért úgy döntöttünk, hogy **mindkét verzió megoldására kitérünk**. Az alap leírás a Spectrum verzió esetében alkalmazható, a C64 gépre készült változat eltéréseit zárójelben, *dőlt betűvel* fogjuk ismertetni. A játéknak egyébként **nem jelent meg Spectrum 128K gépre készült változata**, ezért utólag is elnézést kérünk mindazon Spectrum 128K gép tulajdonosaitól, akik félreértették dícséretünket a program zenéjével kapcsolatban. A leírást nem akartuk bő lére ereszteni, csak a fontos mozzanatokat emeltük ki, a megoldás menetét könnyítik a térkép mellékleten elhelyezett térkép szeletek is, itt annyit meg kell jegyeznünk, hogy a C64 verzióban a 6. szint (THE MANSION) meg lett osztva, az utolsó két szoba 'THE FINAL BATTLE' néven külön szintet képez.

1. szint (CENTRAL PARK):



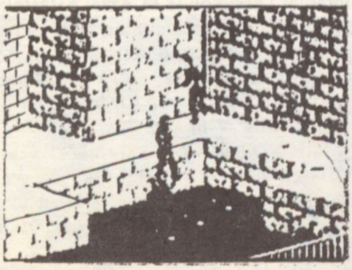
Menjünk a függöny mögé, és hátul törjük be a kis ablakot (belépéskor villog) / menjünk az első szobába, és pottyanjunk le a csapóajtónál / vegyük fel a kulcsot / menjünk ki a szobából / a következő képernyőn induljunk lefelé / a szemetesből vegyük ki a shuriken-t / menjünk át a kapun / kerüljük ki a késdobálót / a következő képernyőn dobjuk le a támadót a shuriken-nel / vegyük fel a térképet / másszunk fel a falon látható rácson / távozzunk balra / ugorjuk át a bemélyedéseket / vegyük fel a botot / ugráljunk visszafelé / másszunk vissza a falról a földre / kerüljük ki a késdobálót / távozzunk jobb-kéz felé / menjünk át a kapun / dobjuk le a támadót a shuriken-nel / vegyük fel a nunchaku felét a női WC-ből / menjünk vissza a kapun és távozzunk felfelé / menjünk be a kapun / dobjuk le a támadót a shuriken-nel / vegyük ki a másik női WC-ből a nunchaku másik felét / menjünk ki és vegyük fel a hot dogot a hot dog standon / menjünk a duplaszárnyú kerítés-kapu közepéhez / válasszuk ki a kulcsot / végezzünk egy felvevő mozdulatot, s a kapu kinyílik / menjünk át a kapun túl / ugorjunk át a folyón a csónak felhasználásával / menjünk a kanyargó ösvényig / álljunk az ösvény végére / fussunk neki és ugorjunk át a szigetre / menjünk a bokrokhoz és a bot segítségével lökjük meg a csónakot / menjünk a sziget alsó széléhez / fussunk neki, és ugorjunk vissza az ösvényre / menjünk tovább az úton, és ugorjunk át a folyón az ellökött csónak felhasználásával / menjünk tovább a park felé és átlépünk a 2.szintre...

2. szint (THE STREET - Az utca):



A parkból kilépve haladjunk lefelé / menjünk át a kereszteződésen, amikor a jelzőlámpa nem villog (*C64 = amikor zöld a jelzőlámpa jelzése*) / a következő képernyőn ismét menjünk át az útkereszteződésen / fussunk az út mentén két képernyőn át / vegyük fel a hamburgert / menjünk tovább és vegyük fel az üveget / menjünk körbe, követve a járda vonalát, egészen a zsákutca végéig / a bódéba menjünk be és végezzünk el egy felvevő mozdulatot, ekkor megszerezzük a csatorna kulcsát / menjünk vissza azon az úton, amerről jöttünk egészen az 'EAT' shop-ig, és menjünk át az úton / kövessük az utat a járda mentén / (*C64 = menjünk át az úton, egészen a zárt ajtóig / rúgjuk be az ajtót / menjünk be a szobába / vegyük fel a kardot / lépjünk ki a szobából / menjünk vissza a sarokig*) / forduljunk el a sarkon / menjünk át az úton / menjünk jobbra, majd az épület után felfelé / vegyük fel a hot dogot / fussunk tovább ugyanebben az irányban a csatornafedélig / a megszerzett kulcsot használva emeljük fel a csatornafedelet / másszunk le a csatornarendszerbe és belépünk a 3.szintre...

3. szint (THE SEWERS - A csatornarendszer):

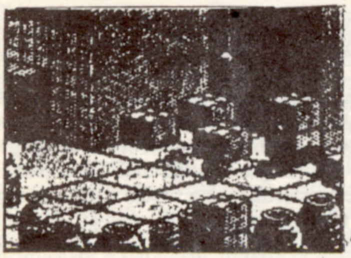


A start pályán induljunk el felfelé / menjünk keresztül a következő képernyőn látható ajtón / szorosan húzódjunk a fal mellé / menjünk be a jobb oldali ajtón / vegyük fel a kulcsot / menjünk vissza / maradjunk szorosan a fal mellett / ugráljunk át a rések felett / (*C64 = várjuk meg amíg leesik az első pók*) / fussunk a következő pályára / keressük meg a csatorna-lejáratot / nyissuk ki a kulccsal / ereszkedjünk le / sétáljunk el a következő pályára / távozzunk a legtávolabbi ajtón (az első kettőn belépve meghalunk) / menjünk a következő képernyőre / lépjünk be a középső ajtón (a két szélsőn belépve meghalunk) / menjünk tovább / a patkányokat ugorjuk át / menjünk tovább, amíg újabb ajtókhoz nem érünk / menjünk be a létrához közelebb eső ajtón / vegyük kezünkbe az üveget / tartsuk a fáklyához, ha meggyulladt, a képernyőn a tárgy-mezőn is látható a változás /

2.1 THE LAST NINJA 2.

menjünk tovább / rakjunk el minden fegyvert (csak az üveg legyen nálunk) / dobjuk rá az aligátorra az üveget / ha elégett, menjünk ki a szemközti ajtón, és belépünk a 4. szintre...

4. szint (THE BASEMENT: Az alagsor):



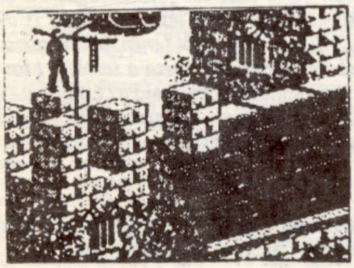
Trükkösen lavírozzunk át a ládák között / a következő pályán másszunk fel a létrán / menjünk vissza az első szintre / vegyük fel a kártyát / menjünk sok-sok képernyőn tovább, amíg egy ajtóhoz nem érünk / menjünk be és vegyük fel a csirkecombot / menjünk vissza a létrához, és másszunk le / menjünk tovább a pályán / amikor elérjük a csilléket, óvatosan menjünk a sínhez olyan közel, ahogy csak lehet / egy csille után gyorsan ugorjunk át / menjünk tovább / az újabb sínt ugorjuk át, mert áram van benne / menjünk tovább / Spectrum = ugráljunk át az oszlopokon egészen az ellenféllel szemköztire / álljunk olyan közel hozzá, hogy elkezdjen küzdeni / győzzük le / (*C64 = az oszlopokon ugráljunk át, megfelelő sorrendben*) / menjünk tovább / menjünk a laboratóriumba / előbb menjünk lefelé / menjünk a heroin porhoz (belépéskor villog) / vegyük kezünkbe a csirkecombot, és érintsük hozzá / menjünk vissza a laboratóriumba / menjünk jobbra / meglátjuk a sarokban a párducot / menjünk közelebb hozzá, amíg fel nem emeli a fejét / (*C64 = amíg fel nem egyenesedik*) / rakjuk le elé a mérgezett csirkecombot / amikor megeszi, holtan esik össze / menjünk tovább / menjünk oda a lifthez és a kártyát tegyük az irányító egységbe / a lift ajtaja kinyílik / belépve az 5. szintre jutunk...

5. szint (THE OFFICE BLOCK - A hivatali blokk):



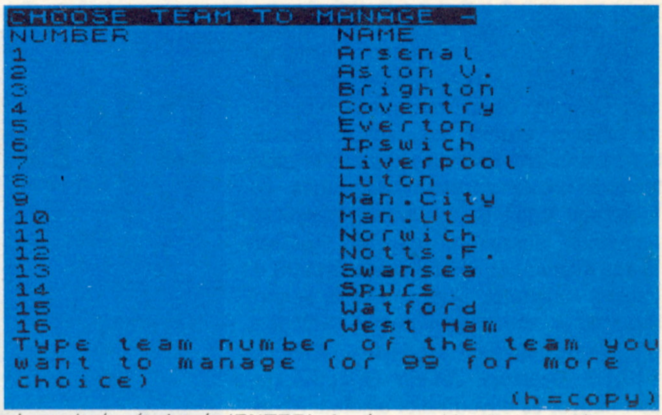
Az első szoba után menjünk tovább az ajtón / ebben a szobában kapcsoljuk be a szemben lévő számítógép terminált, melynek eredményeképpen kapunk egy négyjegyű kódot (ezt jól jegyezzük meg, mert szükségünk lesz rá a 6. szinten) / (*C64 = a kód mindig változik*) / a folyosóra kilépve menjünk tovább / Spectrum = a folyosó végén lépjünk be az első ajtón / vegyük fel a shuriken-t / menjünk ki a szobából / lépjünk le a második ajtón / (*C64 = a folyosó végén lépjünk be a szemközti ajtón*) / nyomjuk meg a jobb oldalon található asztal szélén lévő gombot, ekkor mellettünk kinyílik egy titkos ajtó / menjünk ki a titkos ajtón / másszunk fel a létrán / menjünk be az ajtón / a szoba túloldalán menjünk ki / a ventillátornál nagy ügyességre van szükség / Spectrum = álljunk pontosan szembe a ventillátorral, és miközben csúszunk lefelé, dobjuk meg a shuriken-nel / a ventillátor megáll / (*C64 = nem szabad semmilyen gyors mozdulatot tenni, futni, uugorni, stb. / húzódjunk a falhoz, amennyire csak lehet és apró lépésekkel menjünk át a ventillátor előtt*) / nyissuk ki a rácsos ajtót / menjünk tovább a peremen (vigyázzunk a résekre), amíg egy újabbb létrához nem érünk / másszunk fel rajta / menjünk tovább / meglátjuk a helikoptert, ami ott lebeg a tető mellett / álljunk pontosan a kötélhágcsó elé / fogjuk meg a hágcsót és a helikopter átvisz minket a 6. szintre...

6. szint (THE MANSION - A palota)
C64 gépen + 7. szint (THE FINAL BATTLE - A végső támadás):



Pottyanjunk le a hágcsóról az első oszlopra (a tűzgomb megnyomásával) / ugorjunk oszlopról-oszlopra / ereszkedjünk le a háztetőre / menjünk felfelé / menjünk a peremen a jobb oldali tetőablak mellé / essünk bele az ablakba / menjünk tovább a következő képernyőre / menjünk be az ajtón / vegyük fel a kötelet / menjünk ki a szobából / menjünk vissza az előző képernyőre (ne menjünk le a lépcsőn) / menjünk be az ajtón / menjünk a szemközti nyíláshoz / vegyük a kezünkbe a kötelet és ereszkedjünk le az aknán / a konyhában találjuk magunkat / menjünk ki a konyhából és menjünk be a bal oldali ajtón / a bejárattal szemben a könyvespolc mellett látunk egy kapcsolótáblát / kapcsoljuk ki, ezzel hatástalanítottuk a riasztóberendezést / menjünk ki a szobából / menjünk a virágcserép mögé, ahol egy titkos ajtón bejutunk a pincébe / a kapcsoló pultnál fel tudjuk kapcsolni a következő szoba lámpáját / menjünk tovább / ismét menjünk tovább / a kazánhoz értünk, ahol a gőztől nem tudunk átmenni az ajtón / Spectrum = érintsük meg a kazán első, majd második ajtaját / (*C64 = a kazánon érintsük meg a jobb oldali kapcsolót*) / a gőz a másik irányba kezd fújni / menjünk tovább / (*C64 = itt kezdődik a 7. szint*) / menjünk tovább / az utolsó szobában vagyunk, melynek közepén egy nagy csillag látható / hajtsuk el a szemközti falon lévő festményt / (*C64 = hosszú zászlót*) / egy 'safe' tűnik elő / nyissuk ki a 5. szinten megszerzett kód segítségével / vegyük ki a kristálygömböt, ekkor megjelenik a fő ellenség: KUNITOKI / gyújtsuk meg a gyertyákat / öljük meg KUNITOKI-t / (*C64 = KUNITOKI csak a csillag belsejében ölhető meg, majd ezt követően vissza kell tennünk a kristálygömböt a faliszekrénybe*) / RESET, vagy kapcsoljuk ki a gépet.

2.2 FOOTBALL MANAGER



A meccs a program legérdekesebb része. Igaz, hogy **BASIC**-ban lett megírva, de így is élvezetes látvány. Csak a gólhelyzeteket nézhetjük végig. Természetesen az erősebb csapatnak több ilyen helyzete van, mint a gyengébbnek. A meccs végén a program mégegyszer kiírja az erdményt, valamint a meccsért kapott pénzjutalmat. Ismét '**ENTER**', majd, ha kupamérkőzés volt, a gép kiírja, hogy átjutottunk a következő fordulóba (you're through the next round), vagy kiestünk a kupából (you're out of the cup). Ezután következik a heti költségszámla, ld. később.

Ha bajnoki mérkőzés volt, '**ENTER**' után megkapjuk a többi 14 csapat találkozásának eredményét. Ismét '**ENTER**'. A gép egy kis türelmet kér, majd kiírja a tabellát, amely balról jobbra a következő információkat adja: a csapat neve, az áltata belőtt gólok száma, a kapott gólok száma és a csapat pontja. A pontozás a következőképpen történik: vereség esetén 0 pontot kapunk, döntetlen esetén egyet, győzelemért hármat. Minél több pontja van egy csapatnak, annál előkelőbb helyet foglal el a táblázatban. Egyforma számú pont esetén a gólarány dönti el a helyezést.

Miután megszemléltük a tálázatot, nyomjunk 'ENTER'-t, Ekkor megjelenik a heti költségszámlánk, amely a következő részekből áll:

- játékosok fizetésére költött összeg;
- a pálya bérlésének ára;
- a fizetendő kamat összege (ha van felvett kölcsönünk, annak 1%-át mint kamatot be kell minden héten fizetnünk. **Figyelem!** Ez nem részletfizetést jelent, ettől a tartozásunk változatlan marad. Ha nincs felvett kölcsönünk, ez a pont természetesen 0 font;
- a heti költség egyensúly: ha keresetünk több, mint a kiadásunk, ez a szám pozitív, ha kevesebb, akkor negatív.

Ismét nyomjunk '**ENTER**'-t. Egy újabb fontos részhez értünk el: vásárolhatunk egy játékost. Ezt a lehetőséget azonban - az angol football szabályai szerint - csak akkor kapjuk meg, ha játékosaink száma még nem érte el a 16-ot (*maximálisan 16 játékos lehet egy csapatban*).

A gép kiírja a felkínált játékos képesítését, nevét, erősségét, aktuális energiáját és irányárát. Ekkor be kell írni azt a pénzösszeget, amelyet hajlandóak vagyunk felajánlani érte. Ha nem akarjuk megvenni, írjunk '**99**'-et. Vigyázat, ha megadjuk a játékos árát, még nem biztos, hogy el is fogadják. Ezért, ha feltétlenül meg akarjuk szerezni, kínáljuk érte az irányár kb. 120%-át. Ha szeretünk kockáztatni, felajánlhatunk kevesebbet is, mint amennyit kérnek érte (pl. 70-80%-ot). Ezt is gyakran elfogadják, de az első módszer biztosabb. Ha elfogadták ajánlatunkat, az "*xy has joined your team*" felirat jelenik meg; ha nem, akkor a "*your bid is refused*", majd újabb összeget ajánlhatunk fel, de vigyázzunk, mert az irányára is nő.

Miután véget ért a vásár, ismét a menüt kapjuk meg, és kezdődik a ciklus elölről.

Egy szezon 15 meccsből áll. Ha véget ér, az eredménytábla-beli első három helyezett csapat - nagy pénzjutalom kiséretében - egy ligával feljebb lép. Ha mi is átjutunk, játékosaink erőssége véletlenszerűen megváltozik, áruk pedig megnő.

Végül egy jótanács: csapatunkat meccs előtt annak megfelelően válogassuk össze, amilyen erős az ellenfelünk. Ha az gyenge, "kapcsoljuk" ki az erős - és többnyire megterhelt, kifárasztott - játékosainkat, hogy pihenjenek.

A taktikázáshoz sok sikert, a meccshez sok szerencsét, a játékhoz pedig jó szórakozást kívánunk!

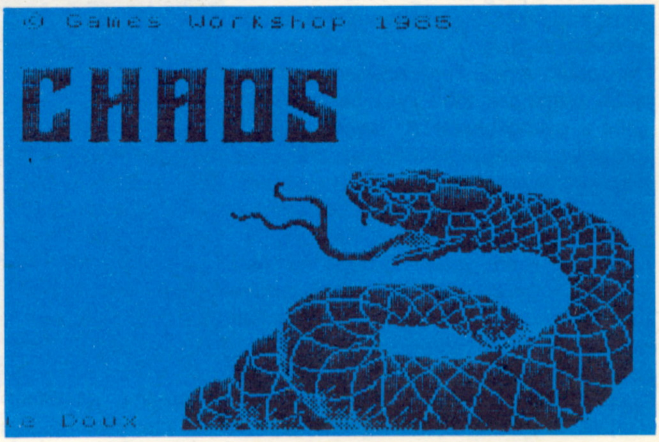
Sir Fred

Nagyon sok visszajelzést kaptunk, hogy a SpV 6. részében közölt örökélet beviteli technika nem hozta meg a hozzá fűzött reményeket, különös tekintettel arra, hogy a file-térképpel is voltak gondok. Most a BASIC/6912/39558/4327/1968 file térképpel rendelkező (MULTIFACE törés) verzióra ismertetjük a bevitel módszerét:

Töltsük be a BASIC loader-t, majd BREAK, és állítsuk le a magnetofont. Írjuk be: LOAD"" SCREEN$: POKE 23838,201: RANDOMIZE USR 23760: POKE 46650,183: RANDOMIZE USR 24833 (ENTER)

Indítsuk tovább a magnetofont, betöltés után végtelen életünk lesz.

2.3 CHAOS



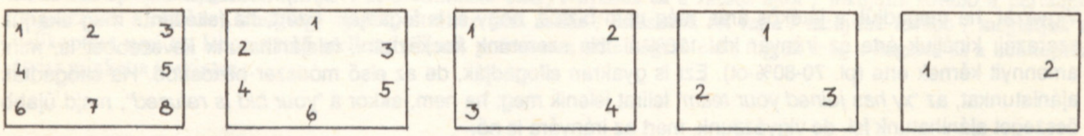
Sok esetben kapunk olyan leveleket a kedves Olvasóktól, melyekben neheztelnek a kiválasztott játékleírásokkal kapcsolatban. Őszintén be kell vallanunk, mi mindig az Olvasók érdekét képviseltük, tesszük ezt most is és a jövőben egyaránt. A játékleírások kiválasztásában elsődleges tényezőnek tekintjük az Olvasói igényeket. Az elmúlt időszakban igen sok levélben kérték tőlünk, hogy közöljünk a **CHAOS** c. játékról ismertetőt. Nos, ami késik az jön, s már itt is van a nagyérdemű, a mindent elsöprő, s egyben csodálatos játék, a **CHAOS**. Szerzője, *JULIEN GOLLOP* igazán nagyot és egyedülállót alkotott. Az ötlet egyedi, a kivitelezés páratlan, így a játék menetét semmihez sem tudjuk hasonlítani, pontosabban ha ezt mégis el kellene követnünk, akkor bátran kijelenthetnénk: *egy sakkprogram és a FEUD keveréke* (?!).

Egy 15x10 osztású táblán folytatjuk játékunkat, célunk elpusztítani a riválís varázslókat, ehhez sok-sok fondorlatra, csalásra, csapdára és persze egy nagy kalap segítségre lesz szükségünk.

A játék három nagyobb egységre tagolható: • kiválasztás, • teremtés, • akció

Ezen túl 32 fordulóból áll, s ez alatt kell a varázslóknak megsemmisíteniük egymást. Érdekesség még, hogy **2-8 játékos is játszhatja**, tetszés szerinti számítógép játékosokat is beleértve. A játék folyamán - ismert emberi szokás szerint - mindenki mindenki ellen van, igaz játszhatjuk úgy is, hogy ketten szövetkezünk pl. 6 számítógép varázsló ellen, vagy akár 4-4 varázslót oszthatunk ki fejenként (2 játékos). Ilyen esetben lehet figyelni, hogy a játékos milyen varázserőt segít elő, s ebből lehet következtetni a teremtményeire, sőt mi az ellenkező erőt növelhetjük, akadályozva ezzel az ellenfelünket (ld. később).

A játék kezdetén ki kell választanunk a varázslók számát (*How many wizards?*) (2-8), majd meg kell adnunk a számítógép által irányított varázslók erejét (*Level of computer wizards?*) (1-8) /nehézségi fok/. Minél nagyobb értéket adunk itt meg, annál nagyobb lesz az összes, számítógép által vezérelt varázsló védekező- és támadó ereje. Az első játékos a bal felső sarokban foglal helyet, a többiek tőle jobbra. A következő ábrákon megmutatjuk a különböző számú játékosok elhelyezkedését (a számok a játékosok számai /1- 1.players/):



Minden varázslóra több jellemzőt kell beállítanunk. Be kell írni a varázslók neveit (max. 12 karakter), majd azt, hogy a varázslót a számítógép, vagy mi irányítjuk (ha a „*computer controlled*" kérdésre 'YES'-szel válaszolunk, a gép fogja irányítani a varázslót, ha 'NO'-val, akkor pedig mi). Ezután kell beállítanunk a varázsló alakját. 8 karakter között választhatunk, ez az erők szempontjából teljesen mellékes, a jó öreg *JULIEN GOLLOP* jóvoltából ugyanis a majdani erőnket véletlenszerűen kapjuk, mindig más lesz a védekező és a támadóerőnk. Ha ez is megtörtént, akkor varázslónk színét választhatjuk ki (8 féle szín közül).
Lényeges: egymás elleni játékban nem szabad egymás varázslatait néznünk!

Ha eddig eljutottunk, akkor a következő menü jelenik meg:

1. **EXAMINE SPELLS**	- Megvizsgálni a varázslatokat
2. **SELECT A SPELL**	- Kiválasztani a varázslatot
3. **EXAMINE BOARD**	- Megvizsgálni a táblát
4. **CONTINUE WITH GAME**	- Folytatni a játékot

Úgy érezzük már annyit papoltunk itt össze-vissza, most már éppen ideje a varázslatokról is egy-két keresetlen szót ejteni. A varázslatokat 4 nagyobb csoportba oszthatjuk, s ezeket ezt követően részletesebben tárgyaljuk:

1. **TEREMTMÉNYEK**
2. **GYENGÍTŐ VARÁZSLATOK**
3. **SEGÍTŐ VARÁZSLATOK**
4. **EGYÉB**

2.3 CHAOS

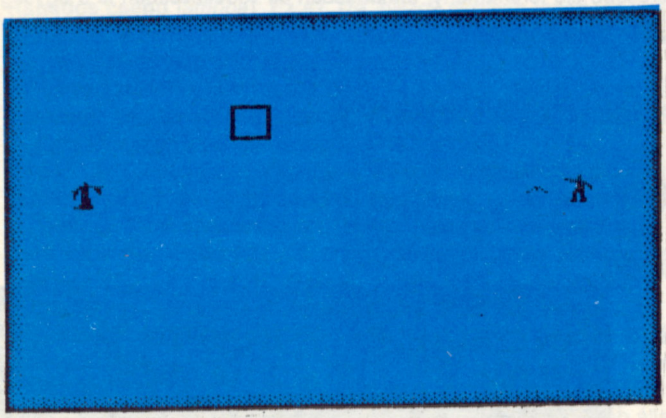
A játék sajnos joystick-kel nem játszható. A billentyűk kiosztása a következő: Q,W,E,A,S,D,Z,X,C (balra fel / fel / jobbra fel / balra / lövés (S) / jobbra / balra le / le / jobbra le). Az 'I' billentyűvel információt kérhetünk a táblán elhelyezkedő bábukról. A billentyűk 1-8-ig a játékosok teremtményeit mutatják meg, a 'K' billentyűvel lehet visszalépni a varázslatból (pl. a teremtésből, ha nincs helyünk, vagy az *'ENGAGED TO ENEMY'*-ből /támadás/).

TEREMTMÉNYEK

A játékos különböző élőlényeket teremthet, melyek egyedi tulajdonságokkal rendelkeznek (pl. a patkány gyenge mind támadásban, mind védekezésben, de sokat tud lépni /hármat/, s így rövid időre képes feltartóztatni egy rivális varázsló bandájának vad támadását). A teremtményekről a főmenüből kérhetünk információt (*EXAMINE SPELLS*), illetve a tábláról az 'I' billentyű segítségével:

combat: Támadás, harci erő.
ranged combat: Messziről jövő támadás erőssége (szó szerint), vagyis pl. ha nyíl van nálunk, az ellőtt nyílvesszővel okozott sérülés nagyságát mutatja meg.
range: Azt jelzi, hogy milyen messzire lőhetünk el.
defence: A védekezési erőt mutatja meg.
movement allowence: Az engedélyezett lépések száma.
manoeuvre raiting: Ez jelzi a manőverezési készséget, azaz pl. ha a készségünk 9 értékű, s 4-et léphetünk, akkor ezt a 4 lépést átlóban is megtehetjük, míg 2 értékű készség esetén csak egyenesen mozoghatunk.
magic resistance: Itt látható a gyengítő varázslatok elleni védekezés mértéke.
spells, ability: Ezek csak a varázslók tulajdonságait mutatják. A *'spells'* a kezdetben fennálló varázslatok számát jelzi, míg az *'ability'* a varázslók különleges képességeiről ad információt (0,1,2).
casting chance: Ez csak akkor jelenik meg, ha a főmenüben vizsgáljuk a teremtményt. A teremtés százalékát jelzi, pontosabban azt, hogy mennyi esélyünk van a teremtmény létrehozására.

A teremtés esélyét (*casting chance*) úgy tudjuk növelni, hogy 2 ellentétes erőt megfelelően növelünk. A két ellentétes erő a **CHAOS-LAW** (*KÁOSZ-TÖRVÉNY*). Ha a főmenüben megvizsgáljuk a teremtményt, a neve mellett megtaláljuk, hogy melyik erő milyen fokozatában kezd el nőni a 'casting chance', vagyis ha a hárpiát megvizsgáljuk, a neve mellett múegtaláljuk, hogy KÁOSZ-1-től nő az esélyünk (*casting chance*). A **KÁOSZ** jele a * (csillag), a **TÖRVÉNYÉ** a ↑ (nyíl). A két erő növelésére varázslattai is van lehetőségünk (*CHAOS 1,2; LAW 1,2*) /ld. később/. Ha kiválasztottuk a megfelelő teremtményt (*select spell*), akkor az illúzió kérdés jelenik meg. Ha egy teremtményt illúziónak teremtünk, az 100 %-ban sikerül, és nem ritka, hogy a valódi lényeknél is erősebb. Nagy hátrányuk viszont, hogy a hitetlenkedés (*disbelive*) varázslattal meg lehet őket semmisíteni. Ha kiválasztunk egy varázslatot, akár sikerül, akár nem, már elhasználtuk (ha sikerül a *'spell succeeds'*, ha nem, akkor a *'spell fails'* felirat jelenik meg). Az egyetlen kivétel a hitetlenkedés, ez soha nem 'fogy el'. A 'teremtmények alfejezet' alatt olyan lényeket értünk, amiket 1 egységre (max.) teremthetünk a varázslótól, azaz a varázsló mellől indulnak.



A táblán egy kis négyzetet irányíthatunk. A sorszámunknak megfelelő számbillentyűvel megvillogtathatjuk eddigi varázslatainkat, és amelyik mozgatható, azzal a következőképpen léphetünk: ráhelyezzük a kurzort, megnyomjuk a tűz-gombot, és a gép kiírja, hogy hány egységet léphetünk. Repülő lényeknél egy újabb kurzort kapunk, és ezzel tudjuk beállítani a repülés célpontját. Ha egy figurát kijelölünk, a számítógép azonnal kijelzi, hogy mi az az 'állat' és kihez tartozik. Ha a *'dead'* nevezetű játékos nincs játékban, akkor ez a jelzés valószínűen azt fogja jelenteni, hogy egy 'állat' hulláját találtuk meg. Ha a lény lába a miénk, ez esetben a tűz-gomb megnyomása után új üzenet jelenik meg: **MOVEMENT RANGE** (*mozgási távolság*), és ezt követően már irányíthatjuk a figurát, a megadott lépések kihasználásával. Ha 3 lépés áll rendelkezésünkre, de ebből csak kettőt akarunk lépni, úgy a '**K**' billentyűvel befejezhetjük a lépést. Repülő állattal csak a célpontot kell kijelölnünk. Lényekkel a támadás nagyon egyszerű - az ellenség közvetlen közelébe kerülve az '**ENGAGED TO ENEMY**' felirat jelenik meg, s ekkor az ellenség irányába mutató billentyűt kell megnyomnunk. Rövid harcot követően a támadott fél vagy elhalálozik (védekezési erejétől függően), vagy életben marad. Ha a teremtmény illúzió volt, a hullája nem marad a helyszínen. Repülő 'kütyükkel' (mint pl. vámpírokkal) a támadás igen kellemes lehet: a kiszemelt

áldozatra irányítjuk a négyzetet (nem mellé), majd megnyomjuk a tűz-gombot. Ha az áldozat átadja lelkét a mindenhatónak, úgy a helyére kerülünk, ha nem, akkor ott maradunk, ahonnan a támadást kezdeményeztük. **FIGYELEM!** Sok kisebb erejű támadás is elpusztíthat egy erősebb lényt (az erősséget a profik előbb-utóbb már fejből tudják, de a kezdők is tájékozódhatnak az 'I' billentyű megnyomásával). *JULIEN GOLLOP* nem csak azzal kalkulálta be a szerencsét a játékba, hogy véletlenszerűen osztja szét a varázslatokat, hanem a teremtmények között is van különbség - 2 azonos teremtmény nem biztos, hogy egyforma erős. A kb. 60 varázslat közül 33 teremtmény, ezeket most megpróbáljuk ,ismertetni, csak azokra térünk ki részletesebben, amelyeket lényegesnek tartunk:

BAT (*denevér*) - Támadása, védekezése a leggyengébb (1-1), viszont 6 egységet tud repülni, kiváló manőverezési készséggel;

BEAR (*medve*) - Nagyon erős állat, 2 egységet tud lépni, a harcban sok hasznát tudjuk venni;

CENTAUR (*kentaur*) - Gyengébbek kedvéért félig ember, félig ló, görög mitológiai alak. Itt kap először szerepet a '*ranged combat*' és a '*range*' is. Az előbbi 4, az utóbbi 2, tehát 4 egység távolságra 2 egység erősségű lényt tud kilőni. Más lényeges tulajdonsága, hogy meg lehet ülni, mint egy lovat, egyébként ezt a neve alatt a '*mount*' felirat is jelzi (mount - lovagolható). Ez gyakorlatilag annyit jelent, hogy ha mellettünk áll a kentaur, az emberrel arra a mezőre kell lépni, ahol a kentaur áll. Így máris rajta 'ülünk'. A támadások csak a kentaurt gyengítik ezután. Ha a többi fordulóban a kentaurral akarunk mozogni, előtte a program meg fogja kérdezni, hogy leszálljon-e a varázslónk a 'lóról' (*Dismount wizard*?);

CROCODILE (*krokodil*) - A neve után látható vízszintes vonal (-) azt jelzi, hogy egyik erőtől sem függ. Itt jegyeznénk meg, hogy amelyik állat a **KÁOSZ**-tól függ, annak a neve után csillag (*), a **TÖRVÉNY**-től függő állat neve után pedig nyíl (↕) található.

DIRE WOLF (*vad farkas*)

ELF (*manó*) - Ez is tud lőni, viszont ha valami lövéskor az útjában áll (a manó és a célpont között), akkor a '*No line of sight*' (nem látod a célt) üzenet jelenik meg.

FAUN (*faun*) - Római istenség, az erdők és nyájak istene. Jellemzője a szarvai és kecske lábai.

EAGLE (sas)

GIANT (*óriás*) - Támadása 9 egység, védekezése 7. Az egyik legerősebb lény, 2 egységnyit léphet.

GIANT RAT (*óriás patkány*)

GHOST (*szellem*) - Halhatatlan teremtmény (*undead*), ez azt jelenti, hogy élő ember nem tudja megölni, csak egy másik halhatatlan, viszont elpusztítható különböző módokon - ld. később. A halhatatlanok között elég gyenge, ugyanakkor tud repülni.

GOBLIN (*kobold*)

GORILLA (*gorilla*)

GOLDEN DRAGON (*aranysárkány*) - A legerősebb élő teremtmény, támadásban és védekezésben is 9 egység erejű. Ezen túl tud repülni, és tüzet is fúj. Sajnos csak 10 %-os, ez a **LAW** növelésével javulhat pozitív irányban;

GREEN DRAGON (*zöld sárkány*)

GRIPHON (*griffmadár*) - Ezt is meg tudjuk lovagolni, és repül is (mount, flying).

HARPY (*hárpia*)

HORSE (*ló*)

HYDRA (*hidra*) - Ez sokfejű görög mitológiai lény.

KING COBRA (*király kobra*)

LION (*oroszlán*)

MANTICORE (*manticore*) - Szintén mitológiai lény, a sárkány és oroszlán keveréke.

OGRE (*emberevő óriás*)

ORC (*harcos*)

PEGASUS (*pegazus*) - A görög mitológia szárnyas lova.

RED DRAGON (*vörös sárkány*)

SKELETON (*csontváz*) - Halhatatlan.

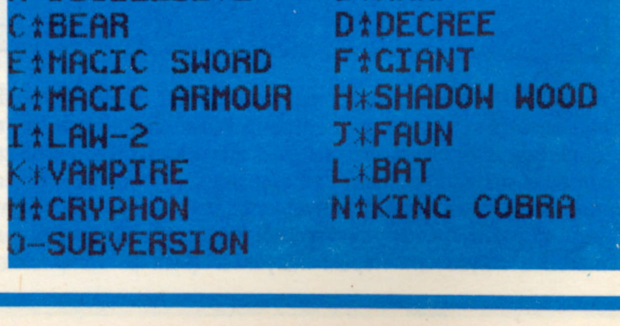
SPECTRE (*kísértet*) - Halhatatlan.

UNICORN (*unikornisz*) - Egyszarvú ló.

VAMPIRE (*vámpír*) - Halhatatlan, repülni képes.

WRAITH (*hasonmás, kísértet*) - Halhatatlan.

ZOMBIE (*zombi*) - Emberevő élő-halott.



GYENGÍTŐ VARÁZSLATOK

Ezekből négyet különböztetünk meg:

- *Dark Power* - sötét erő
- *Justice* - igazság
- *Vengeance* - bosszú
- *Decree* - végzés

Elég hatásos varázslatok, biztos hatást viszont csak kevés varázslat elleni védekezési energiánál tudnak kifejteni. Ha a megtámadott *'magic resistance'*-je alacsony, a hatás nem marad el. A teremtmények elpusztulnak - legyenek azok élők, vagy holtak. Ha viszont varázslókra tesszük rá ezeket, akkor a varázsló nem pusztul el, viszont kedvező hatás esetén, az összes teremtménye felrobban (a gyengítő varázslat hatására úgy legyengül a varázsló, hogy nem tudja életben tartani, és irányítani teremtményeit). A '*Dark power*' és a '*Justice*' által 3 gyengítést tudunk elérni.

SEGÍTŐ VARÁZSLATOK

Ez alatt azt értjük, hogy ezek a varázslatok növelik a varázsló képességeit.
Ide tartoznak:

- Magic Knife - varázskés
- Magic Sword - varázskard
- Magic Shield - varázspajzs
- Magic Armour - varázspáncél
- Magic Bow - varázsnyíl
- Shadow Form - árnyékforma
- Magic Wings - varázsszárnyak

Késsel, karddal és nyíllal minden teremtmény megtámadható. A nyíl nagyon hasznos. A pajzs és a páncél védekezést nyújt. Az árnyékformával hármat léphetünk, végül a szárnyakkal 6 egységnyit repülhetünk.

EGYÉB VARÁZSLATOK

Ebben a csoportban találhatók a legérdekesebb és leghasznosabb varázslatok:

- *Magic Bolt* - gömbvillám
- *Lightning* - villámlás
- *Gooey Blob* - ragadós csepp
- *Wall* - Fal
- *Magic Fire* - varázstűz
- *Chaos 1,2* - káosz 1,2
- *Turmoil* - zűrzavar
- *Dark Citadel* - sötét citadella
- *Subversion* - felforgatás
- *Magic Wood* - varázserdő
- *Magic Castle* - Varázskastély
- *Shadow Wood* - árnyékerdő
- *Law 1,2* - törvény 1,2
- *Raise Dead* - halottat feléleszteni

Gömbvillám: 7 egység távolságra hat, hatásos védekezési mód.

Villámlás: 4 egység távolságra hat, erősebb, mint a gömbvillám.

Ragadós csepp (*amőba*): Ez az 'izé' piszok gyorsan terjed, és elnyel mindent, viszont ha a varázslót éri el, az meghal. Az elnyelt teremtményt ki lehet szabadítani, ha megtámadjuk azt a cseppet, amiben benne van. Egy ilyen amőba nagy kiterjedése végzetes lehet.

Fal: A varázsló köré falat tudunk építeni 4 elemből, viszont a varázslónk mellé ez közvetlenül nem kerülhet.

Varázstűz: Az amőbához hasonló, viszont nem tudjuk elpusztítani, legfeljebb a fal tudja útját állni. Gyorsan terjed, de még a kezdet kezdetén ki is húnyhat.

Káosz 1,2; Törvény 1,2: A Káoszt ill. a Törvényt növelik megadott egységgel (1-gyel, illetve 2-vel).

Zűrzavar: Ezt csak végső esetben használjuk. A legritkább és legtitokzatosabb varázslatok egyike (csak a varázserdőből szerezhető be!). Hatását hiba lenne elárulnunk.

Sötét citadella: Védelmet nyújt, amíg meg tudjuk tartani (sajnos nem örökké tart, kár!). Megtámadhatatlan.

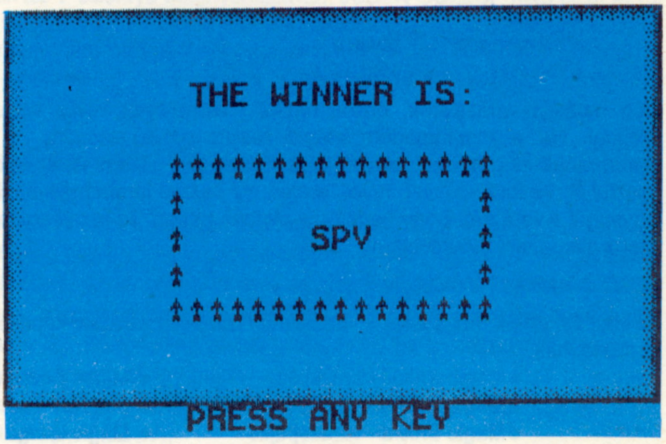
Felforgatás (100 %-ban sikerül): Legjobban a szuggerálás jelző illik rá. Az ellenfél teremtményét megszuggerálva, az a mi irányításunknak fog engedelmeskedni.

Varázserdő: Ez a leghasznosabb varázslat. A számítógép 8 fát helyez el hozzánk minél közelebb (ha kevesebbet, akkor nem volt hely). A fákba be lehet 'menni', azaz (*mint ahogy a Robin of the Wood-ban is láthattuk*) a jóságos fa egy kis idő múlva megajándékoz bennünket egy új varázslattal. Ezt a '*new spell for...*' (új varázslat ...-nak) üzenet is jelzi. **FIGYELEM**! Ló és egyéb állat nem tud fára mászni, így ha rajtuk ülünk, előbb szálljunk le.

Varázskastély: "*Az én váram, az én házam*" - a sötét citadella újabb kiadásban.

Árnyékerdő: Ezt az erdőt már mi 'ültethetjük el' - **FIGYELEM**! Javasolt elhelyezés - kezdjük minél távolabb, és később a közeli fákat telepítsük. Így nem fog megjelenni a '*No line of sight*' (nem látod a fától az erdőt), /vagyis nem látod ázt a helyet, ahova telepíteni akarsz, az előtted álló fa miatt/ üzenet. Ha az ellenség a fák mellé lép, a fa ágaival kioszthatunk jó néhány frászt az ellenség pofázmányaira. A gyengébbek ebbe bele is halhatnak.

Halottat feléleszteni: 4 egység távolságon belül hatásos. A *tetem*-tményre (!) rá kell állítani a négyzetet, és ha szerencsénk van, élő-halott lesz belőle, a tulajdonságai továbbra is megmaradnak (pl. lovagolható), csak éppen már halhatatlanokat is megtámad - egy ilyen sárkány, óriás, medve birtokában gyerekjáték a győzelem kiérdemlése.



ZÁRÓ MEGJEGYZÉSEK

A felforgatás illúziókra nem hat. A varázslókat megvizsgálva láthatjuk, hogy mennyi varázslatuk van, de felhívnánk a figyelmet, hogy ez az érték nem tartalmazza a 'hitetlenkedés'-t, mert az állandó, egyszerű kis varázslat.

A mi általunk eddig elért legnagyobb erő a **KÁOSZ 14** volt, kíváncsiak vagyunk, meg tudja-e ezt dönteni valaki.

Most táblázatos formában ismertetjük a játékban előforduló élőlények egyes tulajdonságait (értékrendben):

Az egyes oszlopok jelentése: 1. lépések száma, 2. támadás, 3. védekezés, 4. a lövés távolsága, 5. a lövés ereje, 6. a támadások elől való kitérés képessége, 7. varázslatok elleni védelem

Élőlény neve	1	2	3	4	5	6	7
Bat (repül)	5	1	1	0	0	9	4
Bear	2	6	7	0	0	6	2
Centaur	4	1	3	4	2	5	5
Crocodile	1	5	6	0	0	2	2
Dire Wolf	3	3	2	0	0	7	2
Eagle	6	3	3	0	0	8	2
Elf	1	1	2	6	2	5	7
Faun	1	3	2	0	0	7	8
Ghost (repül, halhatatlan)	2	1	3	0	0	9	6
Giant	2	9	7	0	0	6	5
Giant Rat	3	1	1	0	0	8	2
Goblin	1	2	4	0	0	4	4
Golden Dragon (repül)	3	9	9	4	5	5	5
Gorilla	1	6	5	0	0	4	2
Green dragon (repül)	3	5	8	6	4	4	4
Gryphoon (repül)	5	3	5	0	0	5	6
Harpy (repül)	5	4	2	0	0	8	5
Horse	4	1	3	0	0	8	1
Hydra	1	7	8	0	0	4	6
King Cobra	1	4	1	0	0	6	1
Lion	4	6	4	0	0	8	3
Manticore (repül)	5	3	6	3	1	6	8
Ogre	1	4	7	0	0	3	6
Orc	1	2	1	0	0	4	4
Pegasus	5	2	4	0	0	6	7
Red Dragon (repül)	3	7	9	5	3	4	5
Skeleton (halhatatlan)	1	3	2	0	0	3	4
Spectre (halhatatlan)	1	4	2	0	0	6	4
Unicorn	4	5	4	0	0	9	7
Vampire (repül)	4	6	8	0	0	6	5
Wraith (halhatatlan)	2	5	5	0	0	4	5
Zombie (halhatatlan)	1	1	1	0	0	2	3

ENTERFACE 10



SPECTRUM programok átírása 1.

Az **ENTERPRISE** gép piacra kerülésekor, és azóta is nagy probléma a programok hiánya (akár játék-, akár felhasználói programról is legyen szó). Ezzel szemben ott van pl. a **SPECTRUM**, amely a maga néhány ezer (!) software termékével a viszonylag jól ellátott gépek közé tartozik. Előbb-utóbb mindenkiben felvetődik a kérdés: hogy lehet a **SPECTRUM** programokat futtatni az **ENTERPRISE** gépen?

A futtatásra több módszer is lehetséges

- A legegyszerűbb venni egy **SPECTRUM-EMULÁTOR**-t. Ennek a módszernek vannak előnyei, de vannak hátrányai is. Előnye, hogy egyszerű, különösebb szakértelmet nem igényel. Hátránya, hogy egyrészt relatíve drága, másrészt nem kezeli a floppy egységet, így aki rendelkezik lemezmeghajtó egységgel, kénytelen minden alkalommal **EMULÁTOR-EXDOS** cserét végrehajtani, ez pedig jelentősen igénybe veszi a gép fólia-érintkezőit, arról nem is beszélve, hogy aki hozzászokott a floppy sebességéhez, az elég nehezen tűri a 'hosszú, méla lest'. Mindemellett a programoknak csak bizonyos százaléka futtatható ezzel a módszerrel.

- Bonyolultabb, de eredményesebb módszer az, amikor a felhasználó fogja a **SPECTRUM** programot, és elmélyült programozással futtathatóvá teszi azt az **ENTERPRISE** gépen. Ennek a módszernek is van hátránya, történetesen az, hogy elég jól kell ismerni a gépi kódú programozást, valamint mindkét gép lelkivilágát (**HARDWARE**), viszont sokkal több előnye van, mint hátránya: bármilyen program átírható ezzel a módszerrel, az átírt programok lemezről is betölthetők, és nem utolsó sorban a program-átíró szellemi épülését is segíti.

Mi a most induló módszertani segédletben a **második módszerrel szeretnénk foglalkozni**, hasznos tanácsokat adva a kedves Olvasónak, azoknak akik elég önbizalommal (és tudással) rendelkeznek ahhoz, hogy egyénileg nekivágjanak egy ilyen mélységű munkának.

Általános áttekintés

Mindenekelőtt azt kell tisztáznunk, hogy mi szükséges egy valamely gépen futó program másik gépre történő átkonvertálásához:

- Gépi kódú program esetén a két gép központi egységének ugyanolyan nyelven kell 'beszélnie', vagy a fogadógépnek legalább szimulálnia kell tudni a forrásgép nyelvét (ettől eltérő esetben is átírható egy program, de a módszer sokkal bonyolultabb). Más (magasszintű) nyelvek esetén nem fontos a kód-kompatibilitás, elég, ha a két fordítóprogram (legyen az akár interpreter, akár compiler) azonos nyelvjárást ismer (Ilyen pl. a CP/M alatt, és az IBM PC-n is futó TURBO-PASCAL is. Az IBM PC-n futó TURBO PASCAL megérti a CP/M alatt futó TURBO PASCAL forrásszövegét, fordítás után futtatható is.) Visszatérve a gépi kódhoz, ez a mi esetünkben biztosítva van, mivel mindkét gép szíve egy Z-80-as mikroprocesszor. Ez azt jelenti, hogy a **SPECTRUM**-ról átvitt kód minden változtatás nélkül futtatható az **ENTERPRISE**-on is, természetesen ez még nem jelenti azt, hogy működni is fog. A működés biztosításához több feltételnek teljesülnie kell.

- Amennyiben a két gép gépi kódja megegyezik, a következő probléma a program átvitele egyik gépről a másikra. Esetünkben a megoldást a két gép magnetofon-illesztőjének hasonlósága adja. Tekintettel arra, hogy a **SPECTRUM** programok nagy része audio kazettán terjed, kézenfekvő az, hogy az átírni kívánt programot magnetofonról olvassuk be.

- Ha a kód már a célgépen található, újabb probléma vetődik fel: a gépek közötti hardware különbség. Ez a mi esetünkben is probléma. Ez az oka annak, hogy programunkat erőteljesen módosítani kell.

- A felépítésbeli különbség maga után vonja az operációs rendszerek különbségét is. A **SPECTRUM** programok esetében elég népszerű a **BASIC-ROM** egyes szubrutinjainak hívása. **ENTERPRISE** gépen ez másképp van megoldva, így más utat kell választanunk. A **ROM** rutinok hívása különösen a felhasználói programok esetében elterjedt. Játékprogramok ritkán hívják meg ezeket a rutinokat, így a játékprogramok átírásához kijelölhetjük a követendő irányvonalat:

1. A program kódját **át kell vinnünk** az **ENTERPRISE** gépre.
2. Az átvitt kódot **hardware szempontból módosítanunk kell**.
3. A módosított programot **el kell látni egy betöltővel**, amely hivatott a programot beolvasni a megfelelő memóriahelyekre, valamint az alapvető hardware emulációt végrehajtani.

Módosítás hardware oldalról

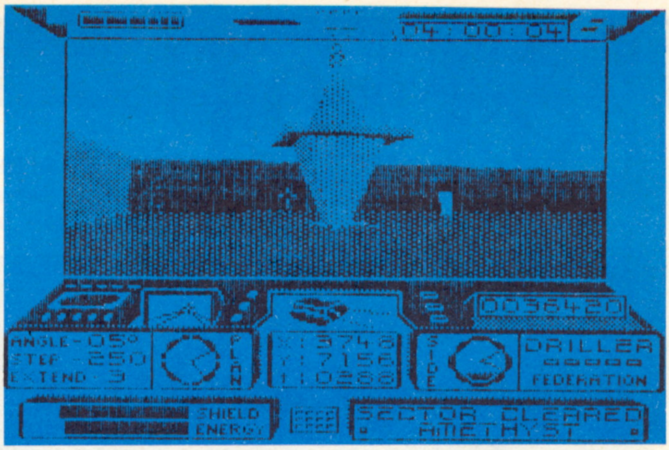
Egy játékprogram esetében három dolgot kell megfelelően átalakítanunk: grafika, hang, irányítás.

Grafika: Készíteni kell egy olyan képernyő üzemmódot, ami megegyezik a **SPECTRUM**-éval. Mint bizonyára minden Olvasó tudja, a **SPECTRUM** ún. attribútum üzemmódban dolgozik. Ez azt jelenti, hogy van egy bittérképe, ahol minden képpontnak egy bit felel meg. Azt, hogy az 1-es ill. 0-ás bit milyen színű legyen, az attribútum byte határozza meg. Nyolc egymás alatti képernyő byte-hoz tartozik egy attribútum byte, vagyis egy attribútum byte-tal egyidőben 8x8 = 64 bit színét állíthatjuk be. A bittérképben lévő 0 biteket háttér, vagy papír színnek (**PAPER**), az 1 biteket pedig kiíró, vagy tinta színnek (**INK**) nevezzük.
Legnagyobb szerencsénkre az **ENTERPRISE** gép video chip-je (**NICK** chip) szintén ismeri az attribútum üzemmódot, igaz egy kicsit másképpen. Ezentúl - hogy ne legyen azért olyan egyszerű a dolgunk - a **SPECTRUM** meglehetősen egyéni módon helyezi el a memóriában a képpont sorokat, valamint speciális a színösszeállítása is.
Készítenünk kell egy olyan **LPT**-t (Line Parameter Table - sorparaméter tábla), amely ilyen felbontással, ilyen színösszeállítással, stb. generál egy **SPECTRUM** képernyőt. A színek miatt később még egy elég nehéz konvertálás is vár ránk.

Hang: A **SPECTRUM** (a normál 48K-s gép) egy egyszerű **BEEP** hangkeltéssel rendelkezik, vagyis egy biten történik a hanggenerálás. Ha a bit 1 értékű, akkor a hangszóró kimozdul, míg ha 0, akkor visszahúzódik, így ha ezt a bitet valaki megfelelő frekvenciával billentgeti, akkor megszólal a gép. Ezt az **ENTERPRISE** gépen elég egyszerű szimulálni, mivel a hanggenerálásért felelős alkatrész (**DAVE** chip) rendelkezik egy un. D/A üzemmóddal. Ez azt jelenti, hogy ilyenkor a hangerőregiszterben megadott nagyságú szint rákerül a hangszóróra, és ott is marad. Gyakorlatilag ez ugyanaz, mint a **SPECTRUM** módszere, de itt nem egy bitet kell billegetni, hanem felváltva 0, és egy 0-nál nagyobb szintet kell a hangerőregiszterbe írni, így még a hangerő is módósítható. Természetesen a programban meg kell keresni az összes hanggenerálással foglalkozó rutint, és a fent leírtaknak megfelelő formára kell hozni.

Irányítás: Ez természetesen már egy extra szolgáltatás, annak van rá szüksége, aki használni is szeretné a programot. A problémát itt is az okozza, hogy a **SPECTRUM** egyéni billentyűzet-figyelést alkalmaz. Itt többféle megoldást is alkalmazhatunk: írhatunk egy általános szubrutint, amely szimulálja az eredeti billentyűzetet, vagy pedig magában a programban kielemezzük a billentyűzet-figyelő rutin működését, és írunk helyette egy másikat. Az utóbbi megoldás elegánsabb, gyorsabb és jobban próbára teszi a programozó képességeit. Az első megoldás kényelmesebb, de alacsonyabb színvonalú.

ENTER FACE



DRILLER

Érdekes, hogy az újabban készülő **ENTERPRISE** - Spectrum átiratok egyre inkább a jobb, újabb programok közül kerülnek ki, és szinte már alig-alig találkozni a *'két lila folt kerget egy pirosat'* kategóriába tartozó játékokkal! Jó példa erre a **DRILLER** is hiszen nemcsak 3D grafikájával, hanem a kemény logikai feladataival is kitűnik a többi közül. Éppen ezért csak a „kezdőlökést" szeretnénk megadni azzal, hogy csak az első pályán segítjük át a játékost!

A játékban sok-sok pályán kell átküzdeni magunkat, fő feladatunk, hogy olajra „lépjünk"! Ehhez a rendelkezésünkre áll egy tank, melynek műszerfalát már a betöltés közben szemügyre vehetjük. Ha betöltődött a játék, válasszuk ki a nekünk megfelelő irányítást:

1. *Keyboard* - 5,6,7,8,0 2. *Sinclair* - 6,7,8,9,0 3. *Kempston* - beépített joy 4. *Cursor* - ld. keyboard

Ha ez megtörtént, nyomjuk meg az '**Ö**' (angol gépen a '**;+**') billentyűt, ugyanis az átirat készítői ezt feleltették meg az '**ENTER**'-nek. Most egy 3D tájképet fogunk látni, bal kéz felől egy fallal. Ez zavarja a kilátást és a városképet. Nyomjuk meg a tűz-gombot. Mint az várható volt, a fal egy része eltűnik, minket persze a maradék is zavar, ezért forduljunk balra, és azt is tűntessük el!

Ezután nyomjuk meg az '**I**' billentyűt, ezzel kérhetünk információt, és ami fontosabb, megáll az óránk, sőt lehetőségünk van a játékállás kimentésére is ('**S**' billentyű), egy elmentett állás betöltésére ('**L**' billentyű), ill. visszaléphetünk a menübe ('**1**'). Mivel most az óránk áll (ez a képernyő jobb felső sarkában ellenőrizhető), bőven jut időnk a műszerek áttanulmányozására.

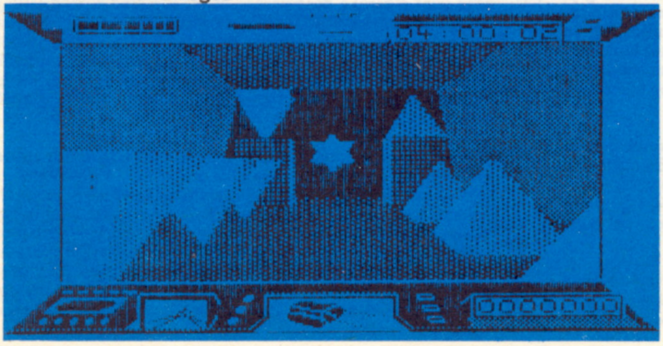
Baloldalon középen látható az *üzemmód jelző*, amely megmutatja, hogy a célkeresztet, vagy a tankot mozgatjuk. Váltani a '**SPACE**'-szel tudunk. Ettől jobbra (középen) az *aktuális járművünk* látható. Később a játék folyamán más járműbe is átszállhatunk. Jobb szélen a *pontszámunkat* olvashatjuk le, melynek a fal eltüntetése következtében kb. 700 körül kell lennie. Az alatta lévő '*SIDE*' feliratú műszeren a gépünk *„bólintási"-szögét* láthatjuk. Ezt a '**P**' és az '**L**' billentyűkkel állíthatjuk. Erre most szükségünk is van, ugyanis a „bunker" mellett álló oszlop tetejét le kell lőnünk. A gépet a vízszintes hossztengelye körül az '**N-M**' billentyűkkel tudjuk forgatni (jobbra-balra billentés). Ettől balra az *X-Y koordinátáinkat*, valamint a *„tengerszint"-feletti magasságunkat* olvashatjuk le. Ettől is balra azt láthatjuk, hogy *milyen irányban* nézünk ki a 'V' két szélesebb szára között. A bal szélen a *fordulás szögét* láthatjuk (állítható az '**A,Y**'- billentyűkkel), alatta az *előre-hátra lépés hosszát* ('**S,X**'), ezalatt pedig a *meghosszabítás mértékét* ('**R,F**') olvashatjuk le. Tulajdonképpen a meghosszabbítással 'emeljük' a gépet. Alatta a *pajzsunk épségét* (*SHIELD*), és az *energiaszintünket* (ENERGY) láthatjuk. Ettől jobbra pedig egy *szöveges kijelzőt* láthatunk.

Valamelyik billentyű lenyomásával lépjünk vissza a játékba. Közelítsük meg a földön lévő fekete keresztet, lehetőleg úgy, hogy a célkereszt a közepére mutasson, majd a '**D**' billentyűvel telepítsük az olajkitermelő egységet. A szöveges kijelzőn leolvashatjuk a termelés mértékét. Ha az elhelyezés nem volt tökéletes, a '**C**' billentyűvel „felszedhetjük", majd ha már beállítottuk, újra telepíthetjük. Ha behatolunk a bunkerbe, ott mind az energiánkat, mind a pajzsunk energiáját növelhetjük, ha a gúlákat szétlőjük. A „talpán" álló a pajzs, a csúcsán álló az energia szintjét növeli

Most néhány jó tanáccsal szeretnénk szolgálni a játékhoz:

- Ne bízzunk meg semmiben, mindenre lőjünk rá, ugyanis így *kapcsolókat, lifteket, ágyúkat* mozgathatunk.
- Mindig próbáljuk az eszünket használni, pl.: mit teszünk, ha erőtérbe kerülünk ? Próbáljuk meg megszüntetni az energiaellátást!
- Nem biztos, hogy egy résen nem férünk át, ellenőrizzük!
- Időnként nézzünk a fejünk fölé!
- Nem minden fal, ami annak látszik!
- Ha már valamilyen eredményre jutottunk, mentsük ki a játék állását, ne kelljen mindig elölről kezdenünk.

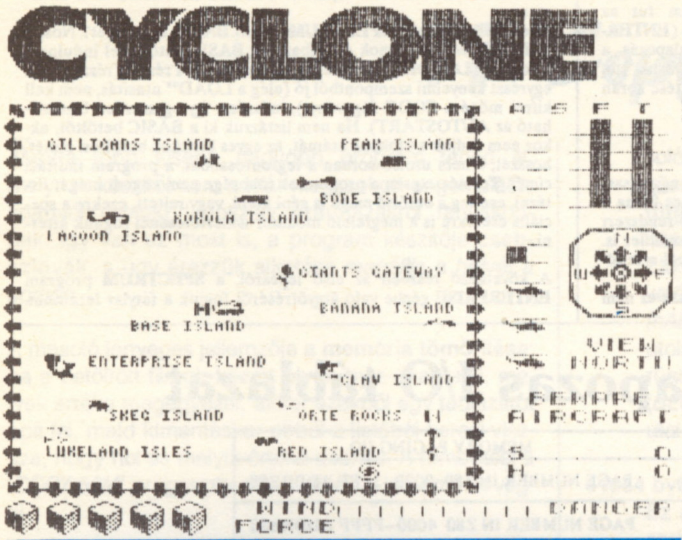
A program eredeti Spectrum változata fut a **hardware emulátor** segítségével is. Az irányítás ebben az esetben is megegyezik az ismertetettel, kivéve az '**ENTER**'-t, mert itt az '**ENTER**' billentyű az '**ENTER**'-t jelenti, és a tüzelés csak a '**0**'-val lehetséges.

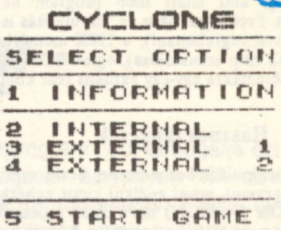


A játékban használatos billentyűk:

A,Z	- szög +/-
S,X	- lépés +/-
F,R	- meghosszabbítás +/-
D,C	- olajkitermelő telepítve /visszahúzva
B	- célkereszt ki/be
SPACE	- célkereszt/gép mozgatása
N,M	- hossztengely irányú forgatás
P,L	- „bólintás" +/-
ENTER,0	- tűz
U	- kitekintés előre/hátra
I	- információ
S	- játékállás kimentése
L	- játékállás betöltése
1	- vissza a menübe







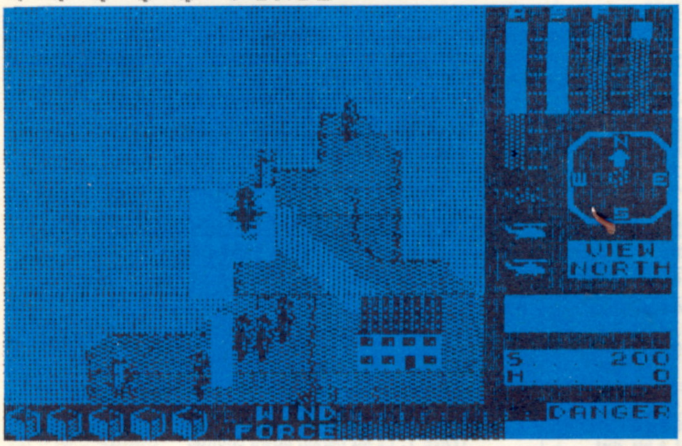
Ez egy régebbi, híres Spectrum játék átirata, eredetileg egyébként a játék szerzője: **COSTA PANAYI** kezei közül került ki a **HIGHWAY ENCOUNTER** c. játék is.

Egy tengeri helikopteres mentőszolgálat pilóta tisztjeként az a feladatunk, hogy összegyűjtsük a szigetekről az embereket, valamint öt ládát. Az embereket és a ládákat vissza kell szállítanunk a légibázisra, miközben menekülni kell a szigetcsoporthoz közeledő ciklon elől.
Dolgunkat nehezítik az időnként feltűnő ellenséges repülőgépek, valamint a ciklon magjához közeledve, annak erősödő szélhatása is.

A képernyő jobb oldalán négy **oszlopgrafikont** láthatunk, ezek a helikopter magassági helyzetét, sebességét, üzemanyagtartalékát, és a mentésre még fordítható időt mutatják. Az ellenséges légierő közeledtére **villogó felírat** figyelmeztet, ilyenkor igyekezzünk minél alacsonyabbra szállni. A ciklontól való távolságunkat a **szélerősségjelző** műszeren követhetjük nyomon. A helikopter és a ciklon helyzetét és mozgását a **térképen** láthatjuk.

A játék indulásakor a menüben beállíthatjuk, hogy beépített joy-t (**INTERNAL**), vagy külső joy-t (**EXTERNAL 1,2**) akarunk-e használni, a játék az '5' billentyű megnyomására indul. A joy használatánál talán furcsa, de hamar meg lehet szokni, hogy felfelé tolva a botkormányt a gép lefelé száll, míg lefelé húzva éppen fordítva. A '**STOP**' billentyűvel megszakíthatjuk a játék futását, az '**M**' billentyűvel a térképet hívhatjuk le, míg a '**JOBB SHIFT**' segítségével a képváltásra nyílik lehetőségünk.

Jó szórakozást, és eredményes mentést kívánunk!



WIZARD'S LAIR

A „**Wizard's lair**" nevű programot a „*Bubble Bus*" software készítette még 1985-ben. A játék kiváló grafikával és zenével rendelkezik. A történet szerint egy varázsló tanyáján - jelen esetben egy piramisban - vagyunk bezárva, ahonnan **csak az arany oroszlán négy darabjának összegyűjtésével juthatunk ki.**
7 szinten kell átküzdenünk magunkat, melyeken az oroszlándarabok véletlenszerűen vannak elszórva. Mi a hetedik szintről indulunk, s a szintek között liftekkel mozoghatunk. Energiánk és lőszerünk állandóan fogytán van, viszont az idővel nem kell takarékoskod-nunk. Mivel az oroszlánok mondig más helyen vannak, ezért az összes pályát be kell járni a játék végigjátszásához. **Cél:** az oroszlándarabok összegyűjtése után elmenni a térképen 'C' betűvel jelzett pályára .

Fontosabb tárgyak és ellenségek:

kulcs, gyémánt, gyűrű:

Ezek segítségével juthatunk be a nem önműködően nyíló ajtókon. Ilyenkor eggyel kevesebb lesz az éppen aktuális tárgyból, s utántöltésük is lehetséges az adott tárgyak felszedésével.

varázslift:

Ebbe a liftbe belelépve a gép ötbetűs nevet kér - jelen esetben annak a pályának a nevét, ahova teleportálni akarunk. Használata akkor célszerű, ha már a négy oroszlándarab nálunk van.

kereszt:

Fevételével lehetővé válik, hogy az egyes kígyókon átjuthassunk - feltétel, hogy a kereszttel megegyező színű legyen.

lőszer:

Lövöldözés közben állandóan fogy, de a lőszer felvételével minden esetben utántölthető. Ahhoz, hogy ne kelljen lőnünk, ezért fel lehet venni egy „izét" - a játékos nevezze úgy, ahogy akarja -, s ilyenkor az ellenségek érintése a halálunkat eredményezi.

A játék megoldását jelentősen megkönnyíti a mellékleten elhelyezett térképek felhasználása. A barangoláshoz sok sikert kívánunk!

Egyéb akadályok

Fejtőrést okozhat több dolog is, pl. az **IM2**-es megszakítás szimulálása. Ez az a dolog, ami miatt több program nem fut az **EMULÁTOR**-on sem. Problémát okoz az '**EI**' utasítás is (**ENTERPRISE**-on másképp kell végrehajtani), a 128K memórialapozás, a **128K** zene (AY-3 8912 chip szimulálása), egyes **ROM hívások**, stb. Az ilyen speciális problémákra egy-egy aktuális rész kifejtése során fogunk kitérni.

Hasznos tanácsok

A program átírási manipulációhoz nem árt beszerezni egy disassembler-monitor programot, mivel enélkül kicsit nehézkes lenne a dolgunk. Mi a **SIMON (ASMON)** nevű Z-80 fejlesztő-rendszert használtuk, ugyanis ez nem csak disassembler, hanem assembler is. Minden - később ismertetésre kerülő - példaprogram ezzel a rendszerrel lett előállítva.

Az sem árt, ha van a közelünkben egy **SPECTRUM**, persze ez nem létszükséglet, csak az eredetivel való egybevetéshez, ill. a BASIC betöltők listázásához kell. Célszerű egy software **SPECTRUM EMULÁTOR** beszerzése is, mivel a BASIC listázására ez is alkalmas.

Miért kell listázni a **SPECTRUM** verzió BASIC betöltőjét? Nos, a **SPECTRUM** programok általában egy BASIC betöltővel indulnak, a betöltő (**LOADER**) tölti be a program további részét (részeit). Ez egyrészt kényelmi szempontból jó (elég a LOAD"" utasítás, nem kell kiírni mögé a CODE függvényt), másrészt így egyszerűen biztosítható az AUTOSTART). Ha nem listázzuk ki a BASIC betöltőt, akkor nem tudjuk a blokkok számát, az egyes blokkok betöltési címét, hosszát, s nem utolsó sorban a legfontosabbat, a program indítási címét. Természetesen a programok többsége nem engedi magát listázni, esetleg a betöltő maga is gépi kódú, vagy rejtett, ezekre a speciális esetekre is a megfelelő módszer ismertetésénél fogunk kitérni.

A következő részben az első lépésről, a **SPECTRUM** program **ENTERPRISE** gépbe való átgyötréséről fogjuk a fátylat lerántani.

Memória lapozás I/O táblázat

MEMORY PAGING I/O		
B0 / 176	IN / OUT	PAGE NUMBER IN Z80 0000–3FFF ADDRESS
B1 / 177	IN / OUT	PAGE NUMBER IN Z80 4000–FFFF ADDRESS
B2 / 178	IN / OUT	PAGE NUMBER IN Z80 8000–BFFF ADDRESS
B3 / 179	IN / OUT	PAGE NUMBER IN Z80 C000–FFFF ADDRESS

Hang-chip I/O táblázat

SOUND I/O								
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
A0	TONE 0: 12 bit PERIOD DOWN COUNTER (LOW)							
A1	RING MODULATION WITH TONE 2:	HIGH PASS CLOCKING WITH TONE 1:	00 Pure Tone 01 4 bit dist. 10 5 bit dist. 11 7 bit dist.		TONE 0: 12 bit period down counter (HIGH)			
A2	TONE 1: 12 bit PERIOD DOWN COUNTER (LOW)							
A3	RING MODULATION WITH NOISE:	HIGH PASS CLOCKING WITH TONE 2:	00 Pure Tone 01 4 bit dist. 10 5 bit dist. 11 7 bit dist.		TONE 1: 12 bit period down counter (HIGH)			
A4	TONE 2: 12 bit PERIOD DOWN COUNTER (LOW)							
A5	RING MODULATION WITH TONE 0:	HIGH PASS CLOCKING WITH NOISE:	00 Pure Tone 01 4 bit dist. 10 5 bit dist. 11 7 bit dist.		TONE 2: 12 bit period down counter (HIGH)			
A6	RING MODULATION WITH TONE 1:	HIGH PASS ON NOISE WITH TONE 0: as CLOCK	LOW PASS ON NOISE WITH TONE 2: as CLOCK	17 and 7 bit counter swap	COUNTER LENGTH SELECT 00 17 bit 10 11 bit 01 15 bit 11 9 bit		NOISE CLOCK SELECT 00 31.25 Khz 10 Tone 1 01 Tone 0 11 Tone 2	
A7	INT SELECT 00 1 Khz 10 Tone 0 01 50 Khz 11 Tone 1			D/A (Right) SELECT	D/A (Left) SELECT	TONE 2: SYNC	TONE 1: SYNC	TONE 0: SYNC
A8			TONE 0: LEFT AMPLITUDE D/A LEFT					
A9			TONE 1: LEFT AMPLITUDE					
AA			TONE 2: LEFT AMPLITUDE					
AB			NOISE: LEFT AMPLITUDE					
AC			TONE 0: RIGHT AMPLITUDE D/A RIGHT					
AD			TONE 1: RIGHT AMPLITUDE					
AE			TONE 2: RIGHT AMPLITUDE					
AF			NOISE: RIGHT AMPLITUDE					

TF-Copy

Egyre több kisebb nagyobb program terjed el, melynek készítői nem a már megszokott angol, olasz, spanyol esetleg jugoszláv származásról tanúskodnak. Így van ez most is, a program készítője **csehszlovák**, s úgy érezzük alkotása ***megállja a helyét a sorban.***

A másoló lényeges jellemzője a **memória tömörítése**: ha a betöltött file-ok között olyan byte-okat talál, melyek értéke megegyezik, akkor ezekből egy táblázatot épít fel, majd kimentéskor ebből a listából keresi vissza, hogy hol és melyik értéke azonos. A tömörítést a ***'COPY 86M'*** *programhoz hasonlóan* a képernyőn végzi.

A program egyaránt fut mind a **48K**-s, mind a **128K**-s géptípusokon. A gépi kód hossza **2816 byte**, kezdőcíme: **62720.**

A TF-COPY funkciói:

CLOCK - Set

Ennek az opciónak a segítségével lehet beállítani, hogy **mekkora szünetet hagyjon a fejléc előtt**, valamint a fejléc és a törzsbyte-ok között. A paraméterek értéke alapértelmezésben: **2400/456.** A változtatás az **'5-8'** billentyűkkel lehetséges, az eredeti értéket az **'N'** billentyűvel állíthatjuk vissza.

MODE - Select

Ezzel állíthatjuk be a COPY helyét, de csak az indítás (*'0' billentyű*) után. A menete a következő: nyomjuk meg az **'M'** billentyűt, majd állítsuk be, hogy a képernyő melyik részét használja fel a program saját célra (*azaz hol tömörítsen*). A megfelelő rész kiválasztása az **'1-3'** billentyűkkel történhet.

RENAMÈ (átnevezés)

Ez az opció igen hasznos, hiszen igen sok másolóprogram nem támogatja ezt a funkciót. Segítségével **bármilyen file-nevet átnevezhetünk** (*természetesen a fejléces file-t*). Használata szintén csak az indítás után lehetséges. Ekkor nyomjuk meg az **'R'** billen-tyűt, ezt követően megjelenik a *RENAME from a TO b (nAB)*, ahol **'a'** jelzi a kezdetet, **'b'** pedig az utolsó átnevezendő file-t. **'n'** esetén visszatérünk, **'A'** (*ALL*) az összes átnevezését jelzi, **'B'** (*BEGIN*) pedig a kezdettől való átnevezésre utal.

41984 byte másolása

Ez a maximális memóriaméret, a tömörítés 14 sorban történik.

44032 byte másolása

Itt a tömörítés 6 sorban történik.

44288 byte másolása

Használja az attributumokat is.

'0' - START

A program indítása.

A TF-COPY hibaüzenetei:

Parity (*többlet*) **byte**: xxxx
Break byte: xxxxx (*annak a byte-nak a címe, ahol a megszakítás történt*).

Ezen kívül a program a már jól megszokott funkciókat ismeri:

L - LOAD, S - SAVE, V - VERIFY, D - DELETE

Mindegyik funkció szintaktikája megegyezik a **RENAME** mellett leírtakkal, azaz

pl. SAVE from a TO b (nAB),
VERIFY from a TO b (nAB), stb.

Pi - R²

Az örökélet beviteléhez a BASIC/6916/20000/20536/1704 file-térképpel rendelkező verzióra van szükségünk.
Töltsük be a loader-t MERGE-dzsel, majd állítsuk le a magnetofont.
Írjuk be a következőket (a sorszám fontos, RESET nem volt!):

```
30 RESTORE 40: FOR i=23823 TO 23835: READ a: POKE i,a: NEXT i
40 DATA 62,255,50,61,150,62,255,55,229,195,86,5
```

Adjuk ki: GO TO 30, ne lepődjünk meg, mert megjelenik az "N Statement Lost" üzenet. Most írjuk be újra a GO TO 30-at, és RANDOMIZE USR 23760-nal indítsuk el a gépi kódú részt. A betöltést folytassuk a 6916-os blokk-tól.

Szótár micro-PROLOG-ban II.

Az előző részben már írtunk egy szótár szavait visszakérdező programot a micro-PROLOG SIMPLE nyelvbővítésében.

Most tekintsük meg és vizsgáljuk meg a működését!

```
LOAD PROLOG

SPECTRUM micro-PROLOG T1.0
(c) 1983 LPA Ltd.
24153 Bytes Free

&.LOAD SIMPLE

&.load VIZSGA

&.list all
a_m teszt if
   X szószám and
   0 LESS X and
   X vizsga
X szószám if
   Y isall (Z : Z a_m x) and
   Y számossága X
() számossága 0
(X|Y) számossága Z if
   Y számossága x and
   SUM (x 1 Z)
X vizsga if
   RND and
   Y RND X and
   SUM (Y 1 Z)
   Z edik x and
   x kérdés and
   X vizsga
X edik Y if
   CL (((a_m Y Z)) X X)
X kérdés if
   P (X = ) and
   Y R and
   X értékelése Y
X értékelése Y if
   X a_m Y
X értékelése Y if
   not X a_m Y and
   X helyesbítése
X helyesbítése if
   PP (Nem jó ! A helyes megoldás : )
   and X válasza
X válasza if
   X a_m Y and
    P and
   Y P and
   FAIL
X válasza if PP
&.
```

Ha összehasonlítjuk az előző részben közölt beviteli listával, észrevehetünk néhány különbséget. Egy-egy x átváltozott y-ná, Z-ből z lett s más változók sem mindig maradtak meg általunk adott formájukban. Ez természetes jelenség, az oka az, hogy a **micro-PROLOG a változókat csak az adott relációban elfoglalt helyük, sorrendjük alapján különbözteti meg.** Ezért egy relációban elsőként szereplő változó mindig X lesz, a második Y, majd a Z,x,y,z,X1,Y1,Z1,x1,y1,z1,...stb., vagyis a nagy és kis x y z -k ellátva sorszámokkal. Ezek és csak ezek jelölhetik a változókat, s a beírás jelölésétől függetlenül ebben a sorrendben tárolódnak a különböző változók.

Tanulmányozzuk a relációkat! Induljunk el felülről lefelé:

a_m teszt

A teszt reláció akkor teljesül, ha argumentuma a_m, s a szótárban levő szavak száma X, ez nagyobb nullánál, s ennyi szóból áll a vizsga. A szószám és a vizsga felhasználói relációk, azaz később definiálni kell őket, míg a LESS rendszerreláció.

X szószám

X a szószám akkor, ha Y azon Z szavak listája, amelyekre teljesül az, hogy van magyar megfelelőjük (azaz benne vannak a szótárban), s az Y listában levő szavak száma X. A számossága reláció definíciója később következik, az isall a SIMPLE relációja. Nagyon hasznos kétargumentumú reláció, akkor igaz, ha az első argumentuma lista és elemei rendre a második argumentum (which formátumú) feltételeit kielégítik. (Megjegyezzük, hogy mint a listakezelésnél mindig, itt is szerepe van a listaelemek sorrendjének, tehát ha másképp rendezzük át őket, akkor már megváltozik a reláció érvényessége.) Az isall-t leggyakrabban - mint most is - az első argumentumot változóként megadva a második argumentumban szereplő feltételeknek eleget tevő elemek listába gyűjtésére használjuk.

X számossága Y

Az üres lista számossága 0. Nem üres lista elemeinek száma megegyezik az első elem elhagyásával keletkező lista számosságának eggyel növelt értékével. A SUM rendszerreláció akkor igaz, ha az első két argumentumának összege megegyezik a harmadik argumentummal. Ellenőrzésre, összeadásra és kivonásra használhatjuk, attól függően, hogy melyik argumentuma szám és melyik változó.

X vizsga

Ez a reláció a teszt lelke. Akkor teljesül, ha a véletlenszám generátor inicializálva van, 0 < = Y < X, véletlen egész az Y, Z eggyel nagyobb nála (tehát 1< = Z< = X), a Z-edik szó x és erre az x-re vonatkozóan kérdés jelenik meg. Az RND argumentum nélkül a véletlenszám generátort inicializálja, két argumentummal pedig az első argumentum (változó) a második argumentumnál kisebb nemnegatív egész értéket vesz fel. A vizsga reláció utolsó feltétele saját magának a teljesülését kívánja meg, ezzel a rekurzióval a tesztelést végtelen ciklusban folytatjuk, amelyből csak BREAK-kel (BREAK = Symbol Shift + SPACE) lehet kiszállni.

X edik Y

A szótárban lévő X-edik angol szó Y, ha a CL rendszerreláció szerint az a_m reláció X -edik kijelentésében szerepel. A CL relációval meg lehet találni egy reláció n. kijelentését, de megfordítva, azt is, hogy egy kijelentés hányadik az adott relációban, attól függően, hogy melyik argumentuma változó.

X kérdés

Az X szó kérdésként szerepel, kinyomtatódik (egyenlőségjellel követve), s beolvasódik Y és X értékelése Y. (Azt, hogy a rendszer bevitelre vár, a szokásos . promt jelzi.)

X értékelése Y

Az X angol szó értékelése Y magyar szó, ha X a_m Y. Egyébként akkor X értékelése Y, ha nem igaz hogy X a_m Y, de ekkor bekövetkezik X helyesbítése.

X helyesbítése

X helyesbítése esetén kiíródik, hogy *Nem jó !* A helyes megoldás : és az X válasza reláció lép be.

X válasza

Az X válasza reláció teljesül(ne), ha X a_m Y (azaz van az adatbázisban megfelelője X -nek), kiíródik egy szóköz, majd Y, s utána letagadja, hogy volt megoldás - ezt a FAIL rendszerreláció okozza, amely sohasem teljesül. Ennek következtében a visszalépéses keresést igénybe véve újabb X a_m Y relációt keres a rendszer, amíg csak ilyen van. Ha már nincs több, akkor az X válasza második kijelentését értékeli ki, amely egy soremelés elvégzésével mindig teljesül. (Ezt a technikát alkalmazza a SIMPLE which relációja is az összes megoldás megkeresésére !)

Néhány megjegyzés a programmal kapcsolatban:

Annak ellenére, hogy rendkívül egyszerű, néhány gyakori buktatót elkerül. Helytelen válasz esetén az összes helyes megoldást felsorolja, s ezek bármelyikét el is fogadja - nem csak a véletlenül kiválasztott szópár magyar felét.

Nem védi ki viszont az inputban azt a lehetőséget, hogy a válasz helyett egy változót (x,y,z stb.) beírva behelyettesítéssel teljesüljön az a_m reláció, azaz helyesnek fogadja el a program a feleletet. A változók inputból való kitiltása ugyan egyszerű, viszont feltételezhető, hogy a saját tanulmányainkat szolgáló programot csak nem akarjuk becsapni.

A program továbbfejlesztése

Felmerülhet az az igény, hogy ne csak angol szavak fordítását gyakoroltathassuk, hanem fordítva, magyar szavak angol megfelelőit és más nyelvű szavakat is ki lehessen kérdezni.

Azt szeretnénk elérni, hogy a tesztet tetszőleges szótárra lehessen alkalmazni, beleértve a közvetítő nyelven alapulókat is.

A szótárreláció nevét a teszt1 programban változó argumentumként használjuk.

&.add(X teszt1 if X szószáma Y and 0 LESS Y and RND and X vizsgája Y)

A teszt1 indítása - ha az összes reláció átírása megtörtént - pl. teszt1 a_m, vagyis egyszerűen a teszt1 után be kell írni a használni kívánt szótár nevét.

Ahhoz, hogy működjön a teszt1 reláció, el kell készíteni a kétargumentumos szószáma és vizsgája relációkat.

&.add(X szószáma Y if Z isall (x : ((X x y)) ?) and Z számossága Y)

Ez lényegében csak az isall SIMPLE reláció argumentumaiban különbözik a szószám relációtól. Itt **két alapvető micro-PROLOG jellegzetességet is tanulmányozhatunk**.

Az egyik a ? reláció. A ? egyargumentumú, argumentuma pedig a micro-PROLOG standard szintakszisának megfelelő formájú feltételek listája. A standard szintakszisban a relációnév áll elől, s azt követik az argumentumok. Pl.:

SIMPLE	standard
x a_m y	(a_m x y)

A ? reláció teljesül, ha az argumentumában szereplő feltételek igazak. A ? a SIMPLE is relációjának megfelelője. Tulajdonképpen erre a relációra épül az egész micro-PROLOG felügyelőprogram.

A másik micro-PROLOG jellegzetesség **a metaváltozók használatának lehetősége**. Ez azt jelenti, hogy változó állhat relációnév helyén is, sőt egész relációt is helyettesíthet. A fenti isall SIMPLE relációban pl. az X helyére futás közben a megfelelő szótár reláció neve kerül :

Z isall (x : ((X x y)) ?)

azt jelenti, hogy Z azon x elemek listája lesz, amelyekre az (X x y) reláció teljesül. Ha X az a_m reláció, akkor az (a_m x y); ha X az m_a, akkor az (m_a x y). (Még egyszer kiemelem,hogy itt (a ? miatt) standard szintakszist kell használni a SIMPLE-ben használt x a_m y vagy az x m_a y stb. helyett!)

A számossága reláción nem kell változtatnunk, az pont ugyanúgy használható itt is mint korábban.

Nézzük meg, mit kell a vizsgán változtatnunk.

&.add(X vizsgája Y if Z RND and SUM (Z 1 x) and (X y z) cl x and X kérdés y and X vizsgája Y)

A lényeges változtatás az x-edik reláció megkeresésében van, mivel már nem alkalmazhatjuk a rendszer CL relációját, hiszen az csak a szótárak kapcsolatait leíró relációkat találja meg, nem pedig az egymással áttételes kapcsolatokon keresztül relációban levő szavakat.
A kérdés reláció mindössze a szótárt meghatározó argumentummal bővült.

A cl relációval tetszőleges feltételeknek eleget tevő objektumok valahanyadikát akarjuk megkeresni. Az inicializálás segítségével felveszünk egy szám relációt, amely a számhoz a nulla értéket rendeli, majd a cl1 relációban addig növeljük a szám reláció argumentumának értékét, míg csak az előírt sorszámú megoldást meg nem találjuk az X metaváltozóval jelzett relációban.

&.add(X cl Y if inicializálás and X cl1 Y)
&.add(inicializálás if szám KILL and ((szám 0)) ADDCL)
&.add(X cl1 Y if (X) ? and ((szám Z)) DELCL and SUM (Z 1 x) and ((szám x)) ADDCL and Y EQ x)

E relációk illusztrálják a PROLOG egyik gyakran nem teljesen megértett tulajdonságát, hogy a változók nem a hagyományos nyelvek logikája szerint használandók. A PROLOG-ban a változó egy objektum, amely a relációk kiértékelése során olyan értékkel helyettesítődik be - ideiglenesem - amelyre a reláció igaz lesz. Ha a további relációk feltételeinek ez az érték már nem tesz eleget, akkor törlődik, és a következő lehetséges behelyettesítésű után folytatódik a relációk kiértékelése. Ez az eljárás alkotja a visszalépéses keresést. Mindezekből következően a programozó értékadással nem módosíthatja a változókat, trükkösen lehet csak egy for jellegű ciklust megírni (erre általában nincs is szükség!), a matematikai műveletek eredményét mindig egy-egy új változó értékeként lehet csak megőrizni.

Az inicializálás reláció teljesül a szám reláció törlésével és a (szám 0) - standard szintakszisú - relációnak az adatbázishoz fűzésével. A cl1 relációban a ? megkeresi az X metaváltozóval jelölt reláció megoldását - ha létezik - s utána a DELCL rendszer utasítással törli a szám relációt, majd ezt eggyel megnövelt argumentummal újra létrehozza (ADDCL), végül megvizsgálja, hogy ez az érték megegyezik-e a kereséshez megadott sorszámmal. Ha igen, akkor a reláció vizsgálata véget ér. Ha nem, akkor a rendszer visszalép, törli az eddigi X-et, s a ? a következő lehetséges megoldást állítja elő. (Ha a megadott sorszám 1-nél kisebb vagy a megoldások számánál nagyobb lenne, akkor a reláció hamissá válna; esetünkben erre nem kerülhet sor a sorszám e korlátok figyelembevételével való generálásából fakadóan.)

Figyeljünk fel arra, hogy a metaváltozók alkalmazhatósága mekkora rugalmasságot biztosít. Az X változó a cl (ill. a cl1)

reláció kiértékelésének kezdetén - ha pl. francia-angol tesztet futtatunk - (f_a Y Z) értékű, s ha a véletlen kegyelméből épp az első, ezt a relációt kielégítő szópárt akarja feladni a program, akkor a reláció teljesül, s az X változó értéke (f_a un one) lesz. Vagyis a változók további változókat tartalmazó kifejezéseket is felvehetnek értékül.

A kérdés relációt alig kell megváltoztatni:

```
&.add(X kérdés Y if P (Y = ) and Z R and értékelés (X Y Z))
```

A szótárra vonatkozó argumentummal bővült saját maga és az értékelés reláció is.

```
&.add(értékelés (X Y Z) if ((X Y Z)) ? and /)
&.add(értékelés (X Y Z) if PP (Nem jó ! A helyes megoldás :) and X
válasz Y)
```

A dupla kiértékelés elkerülhetőségét is láthatjuk ebben a relációban. Az eredeti program mindkét kijelentésében szerepelt az X a_m Y (a másodikban tagadva); a két lehetőség szétválasztása az ún. levágással (/) is megoldható. Ez a reláció mindig teljesül, viszont megakadályozza a további megoldások keresését. Ha tehát az értékelés reláció első kijelentése igaznak bizonyul, akkor már nem kerül sor a második kijelentés megvizsgálására még akkor sem, ha a későbbiek során visszalépés válna szükségessé. (Az első kijelentésben a ? argumentuma standard szintakszisú metaváltozós kifejezés; a kiértékelés során pl. (m_a egy one) alakot vehet fel.)

A válasza' reláció is természetes módon bővült kétargumentumossá.

```
&.add(X válasz Y if ((X Y Z)) ? and " " P and Z P and FAIL)
&.add(X válasz Y if PP)
```

A szerkezete azonos az előző változatéval, csupán a megoldások felsorolását metaváltozós kifejezés - ((X Y Z)) ? - segítségével keresi meg.

Töröljük ki a felesleges relációkat, és nézzük meg a program egészét.

```
&.list all
X teszt1 if
  X szószáma Y and
  0 LESS Y and
  RND and
  X vizsgája Y
X szószáma Y if
  Z isall (x : ((X x y)) ?) and
  Z számossága Y
X vizsgája Y if
  Z RND and
  SUM (Z 1 x) and
  (X y z) cl x and
  X kérdés y and
  X vizsgája Y
X cl Y if
  inicializálás and
  X cl1 Y
inicializálás if
  szám KILL and
  ((szám 0)) ADDCL
X cl1 Y
  if (X) ? and
  ((szám Z)) DELCL and
  SUM (Z 1 x) and
  ((szám x)) ADDCL and
  Y EQ x
X kérdés Y if
  P (Y = ) and
  Z R and értékelés (X Y Z)
értékelés (X Y Z) if
  ((X Y Z)) ? and /
értékelés (X Y Z) if
  PP (Nem jó ! A helyes megoldás :)
  and X válasz Y
X válasz Y if
  ((X Y Z)) ? and
   P and
  Z P and
  FAIL
X válasz Y if
  PP
```

Most már tetszőleges szótárreláció alapján tesztelhetjük mind a magunk, mind mások tudását.

```
&.teszt1 f_a
trois = .three
un = .egy
Nem jó ! A helyes válasz : one a
deux = .two

stb.
```

A fenti program elkészítése már bizonyos jártasságot igényel, különösen a PROLOG filozófiájában, viszont meg is mutatja tömörségét. Kezdőknek inkább egyszerűbb feladatokat megoldó programok írását javasoljuk. A PROLOG lehetővé teszi, hogy mindenki a saját szempontjából fontos kérdéseket készítse el relációként, ezáltal igényei szerint alakíthatja a rendszert.

Írjunk meg egy olyan programot, amelyik egy nyelv szavait kérésünkre lefordítja egy másik nyelvre! Ez nagyon hasznos lehet, ha a szótárt már feltöltöttük, és idegen nyelvű szöveget olvasunk. A Spectrum memóriája már elegendő pl. egy számítástechnikai szakszótár kialakításához.

```
&.add(X fordít if Y R and X válasz Y and X fordít)
```

Próbáljuk ki:

```
&.fordít a_m
one
egy
.two
kettő
.shadow

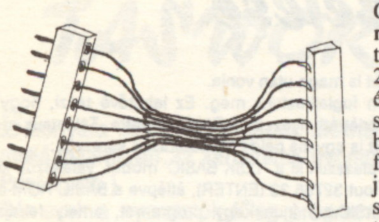
.three
három
.
stb.
```

A végtelen ciklust a BREAK (Symbol Shift + Space) szakítja meg. Ha valamit nem tud lefordítani, akkor semmit sem ír ki és sort emel. A 'válasz' reláció univerzalitásának köszönhetően a 'fordít' is tetszőleges szótár alapján dolgozhat.

Star Raiders II.

A SpV 5. részében már közöltünk egy módszert a STAR RAIDERS II. c. játék örökéletesítésére. Most egy másik file-térképpel elterjedt (BASIC/31397/3987/1968) MULTIFACE-szel megtört verzióhoz ismertetjük a bevitelt:
Töltsük be a BASIC részt, majd BREAK, és írjuk be a következőket: POKE 23838,201: RANDOMIZE USR 23760: POKE 37806,24: POKE 37807,250: RANDOMIZE USR 24833 (ENTER)
Indítsuk tovább a magnetofont, betöltés után örökéletűnk lesz.

1. A fólia megóvása



Gyakori és kellemetlen hiba a billentyűzetfólia törése, szakadása. Ha szerencsénk van, s a kábelrész úgy törik el, hogy marad legalább egy centiméter ép rész, úgy az ábra szerinti megoldást alkalmazhatjuk. Szükségünk lesz 2 db. IC foglalatra, legalább 10 cm hosszú műanyagbevonatú szalagkábelre és egy kis ügyességre is a forrasztás miatt. A rajzon látható módon forrasszuk a kábel egyik végére az IC foglalatot, másikra pedig a kiforrasztott kábel-csatlakozót. Ez utóbbinak a helyére egy másik IC foglalatot célszerű forrasztani. A kábel-csatlakozót illesszük rá a kábel-csonkra, majd myomjuk össze a két IC foglalatot is. Ha lehet, minél alacsonyabb IC foglalatot szerezzünk be, hogy a gép műanyag dobozát minden probléma nélkül össze tudjuk csavarozni!

2. Melegedésgátló

Sokunk problémája, hogy a hálózati feszültségszint ingadozása, magasabb szintje, esetleg gépünk adottsága miatt magas fokú a melegedés (ez sok esetben több kondenzátor idő előtti kiszáradását is eredményezheti). Ha az itt látható áramkört a Spectrum és a hálózati transzformátor közé illesztjük, úgy a melegedést elkerülhetjük.



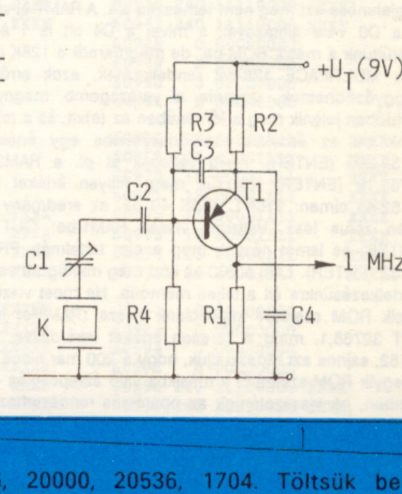
Alkatrészszükséglet:

IC1: uA 7805 stabilizátor
D1 dióda: ZF9 zener
R1 ellenállás

Kondenzátorok:
C1: 150 n típus (3,5V)
C2: 330 ns: 2K
CS1-CS2: hálózati tápforrás dugó/aljzat - O 2.1 mm

3. Módosítás a hanggenerátor áramkörében

Azok között, akik megvették a 'Sinclair Spectrum Játék és Program' c. könyvsoroat 5. kötetét, bizonyára sokan akadtak, akikben szándék ébredt az ott közölt hanggenerátor megépítésére. Nos az IC beszerzésében változatlanul nem tudunk segítséget nyújtani (válaszképpen minden - ezzel a témával foglalkozó - levélíró számára), tudomásunk szerint jelenlegi Müncheni ára: 37,- DM.
Tekintettel az IC magas árára, több változtatással enyhíthetjük kiadásainkat. Pl. a potméterek helyett a régebbi TV-knél, a csatornaváltókban használt 100K-s trimmereket is felhasználhatjuk.
Az órajelet a belső órajel osztása helyett egy tranzisztoros kvarc-oszcillátorral is megvalósíthatjuk. Ez valószínű olcsóbb, mint a több IC segítségével kivitelezett megoldás.



Alkatrészszükséglet:

T1 tranzisztor: BC 212
K kvarckristály: 1 MHz
Kondenzátorok:
C1: 5-30p trimmer
C2: 100p
C3 és C4: 470p
Ellenállások:
R1: 470R
R2: 1,5K
R3: 20K
R4: 33K

Parabola

A program file-térképe: LOADER (241), 6916, 20000, 20536, 1704. Töltsük be a LOADER-t MERGE"" segítségével. Az örökélethez írjuk be a következőket:

```
20 FOR i=23834 TO 23843: READ a: POKE i,a: NEXT i
30 CLEAR 24999: RANDOMIZE USR 23760
40 DATA 175,50,159,149,61,55,229,195,86,5
```

Futtassuk a programot (RUN) és indítsuk el a magnót.



A memória kezelése

A hozzánk érkező levelekben nagyon sokan neheztelnek amiatt, hogy még ezidáig behatóbban nem foglalkoztunk a **128K memória lapozás** módszerével, a teljes memória kihasználása érdekében.

A SpV 7.részében már ismertettük a memóriaszeletek elhelyezkedését, itt kitértünk arra is, hogy melyik port-címmel lehetséges a lapozás végrehajtása, ám úgy tűnik fontos, hogy a módszert néhány **mintapélda** segítségével is illusztráljuk.

Az első félreértést az okozhatta, hogy összekeverhetők a RAM szeletek, a jelölési azonosságok miatt. Nos a felső memóriaterülethez (49152-65535 - C000h-FFFFh) a létező 8 db. RAM szelet mindegyike (RAM0-RAM7) hozzárendelhető, de azt tudnunk kell, hogy a RAM5, mindig a 16384-32767 (4000h-7FFFh) a RAM2 pedig mindig a 32768-49151 (8000h-BFFFh) területhez lesz rendelve, függetlenül attól, hogy azt esetleg fent is kijelöljük. Ez gyakorlatilag azt jelenti, hogy pl. ha a felső memóriaterülethez (49152-65535 - C000h-FFFFh) a RAM2 szeletet rendeljük hozzá, úgy egyidőben két azonos 16K szelet lesz jelen a memóriában.

A lapozásra a 32765-ös port címet tudjuk felhasználni, a port egyes adat bitjeinek értelmezését már megismerhettük a SpV 7. részében. A gyakorlatban mind BASIC, mind gépi kódú oldalról elvégezhető a lapozás.
Kapcsoljuk be a gépet, majd válasszuk ki a '128K BASIC' módot. A port D4-es bitje választja ki, hogy melyik ROM 'éljen', 0 esetén a 128K szerkesztő ROM, míg 1 esetén a hagyományos BASIC-kulcsszó ROM kerül kiválasztásra. Írjuk tehát először is be: out 32765,16 (ENTER). Látszólag semmi nem történt. Ez annak az oka, hogy ha nem változtatunk a RAM-ok felépítésén, úgy ez még nem elegendő a ROM kiválasztásához. Azt is tudjuk, hogy a D5-ös bit a 48K üzemmód kapcsoló, ez könnyen megjegyezhető, ugyanis a D3 (16) + D4 (32) értékét összegezve pontosan 48-at kapunk. Adjuk ki: out 32765,48 (ENTER), és a Copyright felirat már meg is jelenik, 48K-s módban vagyunk.
RESET-eljük a gépet, majd a '128K BASIC' kiválasztása után adjuk ki: out 32765,17 (ENTER). Látszólag ismét a 48K rendszerbe kerültünk, pedig nagy tévedés, mert a Copyright megjelenése ezt még nem támasztja alá. A RAM1-et választottuk ki, a D0 1-be állításával, s mivel a D4 bit is 1 értékű volt, átkerültünk a másik ROM-ba, de megmaradt a 128K rendszer is (akik MULTIFACE 128-cal rendelkeznek, azok erről könnyen meggyőződhetnek, ugyanis a varázsgomb megnyomásakor egyidőben jelenik meg a főmenüben az (e)xit, és a (c)hange is). Írjunk be az érdekes memóriaszeletbe egy értéket: POKE 49152,200 (ENTER), majd váltsunk át pl. a RAM3-ba: OUT 32765,19 (ENTER). Nézzük meg, milyen értéket találunk a 49152-es címen: PRINT PEEK 49152, az eredmény természetesen zérus lesz. Váltsunk vissza RAM1-be: OUT 32765,17 (ENTER) és ismét nézzük meg a cím tartalmát: PRINT PEEK 49152 (ENTER). Lám a 200-as kód még mindig ott van, valóban rendelkezésünkre áll a teljes memória. Ha most viszont csak a másik ROM szeletbe kapcsolunk vissza (RAM1-et megtartva): OUT 32765,1, majd a keresett értéket lekérdezve: print peek 49152, sajnos azt tapasztaljuk, hogy a 200 már nincs ott, vagyis az egyik ROM-szeletből a másikba való átkapcsolás - abban az esetben, ha vissszatérünk az operációs rendszerhez -, a RAM területek törlését is maga után vonja.
A D3 bittel nem foglalkoztunk még. Ez lehetővé teszi, hogy a képernyő-memóriát áthelyezzük a RAM7 szeletre. Tekintsük meg ennek a hasznát is egy kis példa bemutatásán keresztül:
RESET, majd válasszuk ki a '128K BASIC' módot, válasszuk ki a RAM7 szeletet: out 32765,23 (ENTER), átlépve a BASIC ROM-ba. Egyszerűen, BASIC-ből írjunk egy programot, amely felrajzol három kört, és két vonalat a képernyőre, majd ezt a 'bonyolult ábrát áthelyezi a RAM7 szeletre. Itt megjegyeznénk, hogy RAM7-tel azonosítjuk a második képernyő memóriát is.

```
5 BORDER 7: PAPER 7: INK 0: CLS
10 PLOT 0,50: DRAW 255,0
20 PLOT 0,52: DRAW 255,0
30 CIRCLE 50,72,20
40 CIRCLE 90,72,20
50 CIRCLE 130,72,20
60 FOR i=16384 TO 23295: POKE i+32768,PEEK i: NEXT i
```

Most váltsunk át a másik képernyőre: OUT 32765,31 (ENTER), aminek eredményeképpen egy nagy sötétség tárul elénk (ezt kapjuk a MULTIFACE (c)hange parancs aktivizálása esetén is alapállapotban).
'Látatlanban' váltsunk vissza: OUT 32765,23 (ENTER), ismét itt a lista, és futtassuk a BASIC programot: RUN (ENTER).
A futás ideje majdnem másfél perc, ennyi időre van szüksége a BASIC-nek, hogy befejezze az átpakolást. Akinek ez lassú, az lappozzon hátrébb a 'gépi kódú tanfolyamhoz', az ott ismertetett blokk-kezelő utasítások segítségével a művelet a másodperc tört része alatt elvégezhető.

Ha most visszaváltunk a második képernyőre: OUT 32765,31 (ENTER), a képünk itt is megtalálható. Ismét 'látatlanban' váltsunk vissza: OUT 32765,23 (ENTER), majd végezzünk el egy egyszerű trükköt.
Töröljük ki a 60-as sort, a 30., 40., és 50. sort pedig módosítsuk:

```
30 CIRCLE 70,72,20
40 CIRCLE 110,72,20
50 CIRCLE 150,72,20
```

Ezen túl írjuk a programhoz a következő sorokat:

```
60 OUT 32765,31
70 OUT 32765,23
80 GO TO 60
```

Ha futtatjuk, azt tapasztaljuk, hogy a két képernyő váltogatásával animációs hatást érhetünk el.

Egy dolog maradt még hátra, a lapozás gépi kódú utasításainak ismertetése. A RAM7 kiválasztása gépi kódú oldalról a következőképpen lehetséges:

```
30000  01 FD 7F   1,253,127   LD   BC,32765
30003  3E 17      62,23       LD   A,23
30005  ED 79      237,121     OUT  (C),A
30007  C9         201         RET
```

Ezt a sémát már tetszés szerint beépíthetjük saját gépi kódú programjainkba.

Deviants

A SpV 11. részében már ismertettük a játékban megoldható sérthetetlenség beviteli módját. Most az egyes teleport kódokat ismertetjük:

1. CERBURUS, 2. GUZZLOID, 3. ELECTRIX, 4. AURIGIUS, 5. EXCELSOR, 6. XANTHIUS, 7. CYBERNIA, 8. ZACARONT, 9. TROLLDOR, 10. DENIZIENT, 11. DIZZIDIUS, 12. ASIMONUS, 13. VESTRONA, 14. JABBADOR, 15. DEVIANTS

TASWORD-H / Seikosha GP-50

Bizonyára vannak egy páran, akiknek a Spectrum mellé valamilyen 32 karakteres nyomtatójuk van. (ZX *Printer*, stb.)
Bár ezek a nyomtatók főleg programok listázására, kisebb rajzok, táblázatok nyomtatására készültek, tulajdonosaik biztos próbálták már *Tasword*-del szerkesztett szöveg kinyomtatására is használni ezeket. Ennek első próbálkozásra általában egy 64 karakteres, nagyítóval is alig olvasható szöveg szokott lenni az eredménye.
A betűk olvashatóságán javíthat valamit, ha az apróbetűs karakterkészletet átírjuk szögletesebb formára:

```
Bizonyára vannak egy páran, a-      táblázatok nyomtatására készül-
kiknek Spectrumuk mellé valami-     tek, tulajdonosaik biztos pró-
lyen 32 karakteres nyomtatójuk      bálták már Tasword-el szerkesz-
van. / ZX Printer, stb. /           tett szöveg kinyomtatására is
Bár ezek a nyomtatók főleg          használni őket.
programlistázásra, kisebb rajzok
```

Találtunk viszont egy olyan megoldást, amely olvashatóbb szöveget produkál, igaz, hogy csak soronként 32 karakterrel. Ezt szeretnénk most ismertetni.

A dolog a következőképpen kezdődött: véletlenül jöttünk rá arra – a mások által már lehet, hogy közismert dologra –, hogy „ablak" (EXTEND C) üzemmódban nyomtatva a papíron a ROM karakterek jelennek meg, és csak a szöveg bal oldala.

Ebből adódik a megoldás:

1. Csak a képernyő bal oldalára kell írni, a jobb margót 32-re állítva.
2. Nyomtatás előtt át kell váltani ablak üzemmódra.

Ezzel tulajdonképpen a dolog el is lenne intézve, ha a *Tasword H* ékezetes karakterei így is megjelennének. Sajnos eszük ágában sincs. 32 karakteres üzemmódban a magyar betűk helyett a ROM-beli megfelelőjük jelenik meg. (á helyett &, é helyett ', stb.)
A *„Spectrum játék és program"* c. könyvsorozat IV. kötetében (LSI ATSz. 1987) is hivatkoztunk rá, hogy van 16 széles karakter is, amit a *Tasword* író1 a RAM-ban helyeztek el. Ezek a „komolyabb nyomtatókat vezérlő grafikus karakterek", számuk 16 darab.

Sajnos ezek a karakterek a *ZX printert* nem vezérlik, hanem megjelennek a nyomtatott szövegben úgy 64, mint 32 karakteres üzemmódban, vagyis, ha ezeket a karaktereket átdefiniáljuk ékezetes betűkké, jó esélyünk van arra, hogy célhoz érünk.
A nyomtató-vezérlő karakterek a *Tasword*-ben a dec. 60928-61055 címek között helyezkednek el a következő sorrendben:
8,1,2,3,4,5,6,7 *7,*6,*5,*4,*3,*2,*1, ahol a '*' jel az inverz (Shift-elt) karaktert jelenti:

60928	255	11111111	XXXXXXXX	60936	15	00001111	XXXX	60944	240	11110000	XXXX
60929	129	10000001	X X	60937	15	00001111	XXXX	60945	240	11110000	XXXX
60930	129	10000001	X X	60938	15	00001111	XXXX	60946	240	11110000	XXXX
60931	129	10000001	X X	60939	15	00001111	XXXX	60947	240	11110000	XXXX
60932	129	10000001	X X	60940	0	00000000		60948	0	00000000	
60933	129	10000001	X X	60941	0	00000000		60949	0	00000000	
60934	129	10000001	X X	60942	0	00000000		60950	0	00000000	
60935	255	11111111	XXXXXXXX	60943	0	00000000		60951	0	00000000	
60952	255	11111111	XXXXXXXX	60960	0	00000000		60968	15	00001111	XXXX
60953	255	11111111	XXXXXXXX	60961	0	00000000		60969	15	00001111	XXXX
60954	255	11111111	XXXXXXXX	60962	0	00000000		60970	15	00001111	XXXX
60955	255	11111111	XXXXXXXX	60963	0	00000000		60971	15	00001111	XXXX
60956	0	00000000		60964	15	00001111	XXXX	60972	15	00001111	XXXX
60957	0	00000000		60965	15	00001111	XXXX	60973	15	00001111	XXXX
60958	0	00000000		60966	15	00001111	XXXX	60974	15	00001111	XXXX
60959	0	00000000		60967	15	00001111	XXXX	60975	15	00001111	XXXX
60976	240	11110000	XXXX	60984	255	11111111	XXXXXXXX	60992	0	00000000	
60977	240	11110000	XXXX	60985	255	11111111	XXXXXXXX	60993	0	00000000	
60978	240	11110000	XXXX	60986	255	11111111	XXXXXXXX	60994	0	00000000	
60979	240	11110000	XXXX	60987	255	11111111	XXXXXXXX	60995	0	00000000	
60980	15	00001111	XXXX	60988	15	00001111	XXXX	60996	240	11110000	XXXX
60981	15	00001111	XXXX	60989	15	00001111	XXXX	60997	240	11110000	XXXX
60982	15	00001111	XXXX	60990	15	00001111	XXXX	60998	240	11110000	XXXX
60983	15	00001111	XXXX	60991	15	00001111	XXXX	60999	240	11110000	XXXX
61000	15	00001111	XXXX	61008	240	11110000	XXXX	61016	255	11111111	XXXXXXXX
61001	15	00001111	XXXX	61009	240	11110000	XXXX	61017	255	11111111	XXXXXXXX
61002	15	00001111	XXXX	61010	240	11110000	XXXX	61018	255	11111111	XXXXXXXX
61003	15	00001111	XXXX	61011	240	11110000	XXXX	61019	255	11111111	XXXXXXXX
61004	240	11110000	XXXX	61012	240	11110000	XXXX	61020	240	11110000	XXXX
61005	240	11110000	XXXX	61013	240	11110000	XXXX	61021	240	11110000	XXXX
61006	240	11110000	XXXX	61014	240	11110000	XXXX	61022	240	11110000	XXXX
61007	240	11110000	XXXX	61015	240	11110000	XXXX	61023	240	11110000	XXXX

```
61024   0 00000000              61032  15 00001111     XXXX
61025   0 00000000              61033  15 00001111     XXXX
61026   0 00000000              61034  15 00001111     XXXX
61027   0 00000000              61035  15 00001111     XXXX
61028 255 1111111 XXXXXXXX      61036 255 1111111 XXXXXXXX
61029 255 1111111 XXXXXXXX      61037 255 1111111 XXXXXXXX
61030 255 1111111 XXXXXXXX      61038 255 1111111 XXXXXXXX
61031 255 1111111 XXXXXXXX      61039 255 1111111 XXXXXXXX

61040 240 11110000 XXXX         61048 255 1111111 XXXXXXXX
61041 240 11110000 XXXX         61049 255 1111111 XXXXXXXX
61042 240 11110000 XXXX         61050 255 1111111 XXXXXXXX
61043 240 11110000 XXXX         61051 255 1111111 XXXXXXXX
61044 255 1111111 XXXXXXXX      61052 255 1111111 XXXXXXXX
61045 255 1111111 XXXXXXXX      61053 255 1111111 XXXXXXXX
61046 255 1111111 XXXXXXXX      61054 255 1111111 XXXXXXXX
61047 255 1111111 XXXXXXXX      61055 255 1111111 XXXXXXXX
```

A karaktereket a következő lista szerint átdefiniálhatjuk:

```
60928 255 1111111 XXXXXXXX   60936   4 00000100      X       60944   4 00000100      X
60929 129 10000001 X      X  60937   8 00001000     X        60945   8 00001000     X
60930 129 10000001 X      X  60938  56 00111000   XXX        60946  56 00111000   XXX
60931 129 10000001 X      X  60939   4 00000100      X       60947  68 01000100  X   X
60932 129 10000001 X      X  60940  60 00111100   XXXX       60948 120 01111000  XXXX
60933 129 10000001 X      X  60941  68 01000100  X   X       60949  64 01000000  X
60934 129 10000001 X      X  60942  60 00111100   XXXX       60950  60 00111100   XXXX
60935 255 1111111 XXXXXXXX   60943   0 00000000              60951   0 00000000

60952   8 00001000     X     60960   4 00000100      X       60968  40 00101000   X X
60953  16 00010000    X      60961   8 00001000     X        60969   0 00000000
60954   0 00000000           60962  56 00111000   XXX        60970  56 00111000   XXX
60955  48 00110000   XX      60963  68 01000100  X   X       60971  68 01000100  X   X
60956  16 00010000    X      60964  68 01000100  X   X       60972  68 01000100  X   X
60957  16 00010000    X      60965  68 01000100  X   X       60973  68 01000100  X   X
60958  56 00111000   XXX     60966  56 00111000   XXX        60974  56 00111000   XXX
60959   0 00000000           60967   0 00000000              60975   0 00000000

60976   8 00001000     X     60984  36 00100100   X  X       60992  36 00100100   X  X
60977  16 00010000    X      60985   0 00000000              60993   0 00000000
60978  68 01000100  X   X    60986  68 01000100  X   X       60994  66 01000010  X    X
60979  68 01000100  X   X    60987  68 01000100  X   X       60995  66 01000010  X    X
60980  68 01000100  X   X    60988  68 01000100  X   X       60996  66 01000010  X    X
60981  68 01000100  X   X    60989  68 01000100  X   X       60997  66 01000010  X    X
60982  56 00111000   XXX     60990  56 00111000   XXX        60998  60 00111100   XXXX
60983   0 00000000           60991   0 00000000              60999   0 00000000

61000   4 00000100      X    61008  36 00100100   X  X       61016   4 00000100      X
61001   8 00001000     X     61009   0 00000000              61017   8 00001000     X
61002  66 01000010  X        61010  60 00111100   XXXX       61018  60 00111100   XXXX
61003  66 01000010  X        61011  66 01000010  X    X      61019  66 01000010  X    X
61004  66 01000010  X        61012  66 01000010  X    X      61020  66 01000010  X    X
61005  66 01000010  X        61013  66 01000010  X    X      61021  66 01000010  X    X
61006  60 00111100   XXXX    61014  60 00111100   XXXX       61022  60 00111100   XXXX
61007   0 00000000           61015   0 00000000              61023   0 00000000
```

```
61024   0 00000000              61032   4 00000100      X
61025   0 00000000              61033   8 00001000     X
61026   0 00000000              61034 126 01111110  XXXXXX
61027   0 00000000              61035  64 01000000  X
61028 255 11111111 XXXXXXXX     61036 126 01111110  XXXXXX
61029 255 11111111 XXXXXXXX     61037  64 01000000  X
61030 255 11111111 XXXXXXXX     61038 126 01111110  XXXXXX
61031 255 11111111 XXXXXXXX     61039   0 00000000

61040   4 00000100      X       61048 255 11111111 XXXXXXXX
61041   8 00001000     X        61049 255 11111111 XXXXXXXX
61042  60 00111100   XXXX       61050 255 11111111 XXXXXXXX
61043  66 01000010  X    X      61051 255 11111111 XXXXXXXX
61044 126 01111110  XXXXXX      61052 255 11111111 XXXXXXXX
61045  66 01000010  X    X      61053 255 11111111 XXXXXXXX
61046  66 01000010  X    X      61054 255 11111111 XXXXXXXX
61047   0 00000000              61055 255 11111111 XXXXXXXX
```

Mivel a Tasword használata 64 karakteres üzemmódban a legkényelmesebb, és hogy ne végezzünk félmunkát, írjuk át a keskeny grafikus karaktereket is a következő bit-térkép segítségével:

```
61952  15 00001111     XXXX    61960   1 00000001        X     61968   1 00000001        X
61953   9 00001001     X  X    61961   2 00000010       X      61969   2 00000010       X
61954   9 00001001     X  X    61962   7 00000111      XXX     61970   7 00000111      XXX
61955   9 00001001     X  X    61963   1 00000001        X     61971   5 00000101      X X
61956   9 00001001     X  X    61964   7 00000111      XXX     61972   7 00000111      XXX
61957   9 00001001     X  X    61965   5 00000101      X X     61973   4 00000100      X
61958   9 00001001     X  X    61966   7 00000111      XXX     61974   7 00000111      XXX
61959  15 00001111     XXXX    61967   0 00000000              61975   0 00000000
```

61976	1	00000001	X	61984	1	00000001	X	61992	5	00000101	X X
61977	2	00000010	X	61985	2	00000010	X	61993	2	00000010	X
61978	0	00000000		61986	5	00000101	X X	61994	5	00000101	X X
61979	6	00000110	XX	61987	5	00000101	X X	61995	5	00000101	X X
61980	2	00000010	X	61988	5	00000101	X X	61996	5	00000101	X X
61981	2	00000010	X	61989	5	00000101	X X	61997	5	00000101	X X
61982	7	00000111	XXX	61990	2	00000010	X	61998	2	00000010	X
61983	0	00000000		61991	0	00000000		61999	0	00000000	
62000	1	00000001	X	62008	0	00000000		62016	5	00000101	X X
62001	2	00000010	X	62009	5	00000101	X X	62017	0	00000000	
62002	0	00000000		62010	0	00000000		62018	5	00000101	X X
62003	5	00000101	X X	62011	5	00000101	X X	62019	5	00000101	X X
62004	5	00000101	X X	62012	5	00000101	X X	62020	5	00000101	X X
62005	5	00000101	X X	62013	5	00000101	X X	62021	5	00000101	X X
62006	7	00000111	XXX	62014	7	00000111	XXX	62022	7	00000111	XXX
62007	0	00000000		62015	0	00000000		62023	0	00000000	
62024	2	00000010	X	62032	5	00000101	X X	62040	1	00000001	X
62025	0	00000000		62033	0	00000000		62041	2	00000010	X
62026	5	00000101	X	62034	7	00000111	XXX	62042	7	00000111	XXXX
62027	5	00000101	X	62035	5	00000101	X X	62043	5	00000101	X XX
62028	5	00000101	X	62036	5	00000101	X X	62044	5	00000101	X XX
62029	5	00000101	X	62037	5	00000101	X X	62045	5	00000101	X XX
62030	7	00000111	XX	62038	7	00000111	XXX	62046	7	00000111	XXXX
62031	0	00000000		62039	0	00000000		62047	0	00000000	

62048	0	00000000		62056	2	00000010	X
62049	0	00000000		62057	7	00000111	XXX
62050	0	00000000		62058	4	00000100	X
62051	0	00000000		62059	6	00000110	XX
62052	15	00001111	XXXX	62060	4	00000100	X
62053	15	00001111	XXXX	62061	4	00000100	X
62054	15	00001111	XXXX	62062	7	00000111	XXX
62055	15	00001111	XXXX	62063	0	00000000	
62064	2	00000010	X	62072	15	00001111	XXXX
62065	7	00000111	XXX	62073	15	00001111	XXXX
62066	5	00000101	X X	62074	15	00001111	XXXX
62067	5	00000101	X X	62075	15	00001111	XXXX
62068	7	00000111	XXX	62076	15	00001111	XXXX
62069	5	00000101	X X	62077	15	00001111	XXXX
62070	5	00000101	X X	62078	15	00001111	XXXX
62071	0	00000000		62079	15	00001111	XXXX

Ezzel a Tasword átalakítását be is fejeztük.

Az átalakított Tasword

Az így átalakított programon az ékezetes betűk grafikus üzemmódban érhetők el a következő kiosztás-ban:

1 - á 2 - é 3 - í 4 - ó 5 - ö 6 - ú 7 - ü

CAPS SHIFT-tel

1 - Á 2 - É 3 - ▬ 4 - Ó 5 - Ö 6 - Ú 7 - Ü

Ez nem jelenti azt, hogy ha ékezetes betűre van szükségünk, akkor állandóan váltogatni kell a grafikus és normál üzemmód között. Szövegírás közben a Tasword nyugodtan maradhat állandó grafikus üzemben, így a magyar kisbetűk '**Shift**' nélkül bármikor elérhetők. Visszaváltani csak akkor kell, ha számokat akarunk írni, vagy valamilyen szerkesztési funkciót akarunk igénybe venni, pl.: kurzornyilak, CAPS LOCK, TRUE- INV. VIDEO, EDIT (mankó), tehát azok a funkciók, amelyek valamely számbillentyű lenyomására aktivizálódnak. Van még egy előnyünk az új változattal: 16 szabad helyünk van, így több ékezetes betűt definiálhatunk, mint amennyi az alap **Tasword** H-ban van. Ezek a következők: í ó ú Ó Ú.

A Spectrum 8x8 képpont méretű betű-mátrixa miatt nem definiálható a hosszú nagy Í, a hosszú ő és ű.

Végül a hátrányokról:

Ezzel a módszerrel sajnos a Tasword-be írható szöveg hossza a felére csökken. A beírt szöveget nehezebb áttekinteni, mintha az egész képernyőre írtunk volna.

A formátum így sem A4-es, és a lapdobásról is magunknak kell valamilyen formában gondoskodni, bár ezek inkább már a nyomtató hibái.

Mindenesetre ez is egy lehetőség, ami nem jár sok munkával, ezért érdemes kipróbálni.

A karakterek átírása:

Ez gondoljuk senkinek nem fog gondot okozni. Mi a Beta Basic 1.8 program felhasználását javasoljuk, a következő lista szerint:

```
 5 LET c=21184
10 FOR n=40000 TO 41000 STEP 8
15 LET o=n: LET p=n+c
20 FOR m=1 TO 8
22 REM INPUT (p),b: POKE o,b
25 LET a$=BIN$(PEEK o)
30 LET b$="XXXXXXXX"
35 PRINT TAB 0;p;TAB 7;PEEK o;
TAB 12;a$;
40 FOR k=1 TO 8
45 IF a$(k)="0" THEN LET b$(k)
=" "
50 NEXT k
55 PRINT TAB 22;b$
60 LET o=o+1: LET p=p+1
65 NEXT m
70 PRINT
75 NEXT n
```

Miután a karakterkészletet magnóra mentettük, a 40000-es címre visszatöltöttük.

Ezzel a programmal a karakterek kinagyított képét láthatjuk, ha a 22-es sorból elhagyjuk a REM utasítást, az átdefiniálást is elvégezhetjük vele.

Tudjuk, hogy elég speciális témakört választottunk most ki a 'programozástechnika' c. részhez, de az Olvasók által beküldött levelek alapján azt kellett megállapítanunk, hogy sokan rendelkeznek ilyen, vagy hasonló típusú nyomtatóval, ezért talán **nem haszontalan** mindaz, amit most közzétettünk.

Ground Attack

Az örökélet beviteléhez töltsük be a BASIC loader-t, majd BREAK, és állítsuk le a magnetofont. Írjuk be közvetlenül a RANDOMIZE USR 27620 utasítás elé: POKE 29063,0:..., majd ENTER, RUN (ENTER) és indítsuk tovább a magnetofont.

XII. Blokk-kezelő utasítások

A Z-80 gépi kódú utasításai között igen előkelő szerepet töltenek be ezek az utasítások. Gyakran szükségünk lehet egy meghatározott memóriaterület együttes mozgatására, esetleg egy meghatározott memóriaterületen belül adat keresésére. Ezzel már el is árultuk, hogy valójában mire használatosak ezek az utasítások.

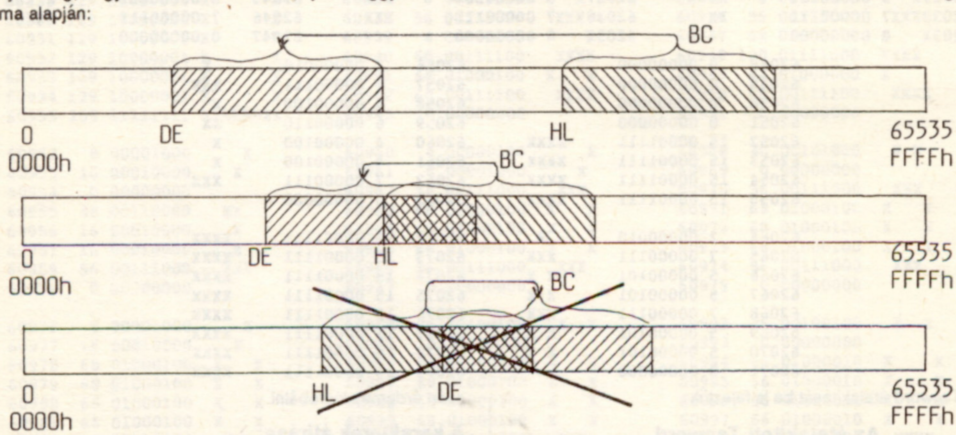
Minden blokk-kezelő utasítás működésének alapszabálya, hogy az utasítás végrehajtását megelőzően a kiválasztott memóriaterület báziscímét a 'HL' regiszterpárban, a memóriaterület nagyságát (a byte-ok számát) a 'BC' regiszterpárban kell elhelyezni. Ha adatblokkot mozgatunk, ez esetben meg kell adnunk a célhely báziscímét is, ezt a 'DE' regiszterpárban állítjuk be. Adatkeresés esetén pedig az 'A' regiszterben kell megadnunk a keresett adat kódját. Minden további kommentár nélkül tekintsük át sorban az ide tartozó 8 db. utasítást - Olvasói kérésre - mintapéldákkal és ábrákkal illusztrálva.

1. Automatikus blokk-kezelő utasítások

Az automatikus blokk-kezelő utasítások jellemzője, hogy mind a mozgatás, mind a keresés **teljesen automatikus**, vagyis a léptetés automatikusan végrehajtódik, minden különösebb programozói beavatkozás nélkül. Éppen ezért ezeknek az utasításoknak nincs kötött ciklusidejük, ugyanis a végrehajtási idő az áthelyezendő, vagy vizsgálandó byte-ok számától függ.

a) LDIR utasítás

Úgy ítéljük meg, hogy ezt az utasítást használjuk a leggyakrabban blokk-kezelő funkcióra. Az utasítás működését tekintsük meg néhány elvi séma alapján:



Jól láthatjuk, hogy az átmozgatandó blokk kezdőcímét a 'HL' regiszterpárban, a célhely címét a 'DE' regiszterpárban, míg az átmozgatandó byte-ok számát a 'BC' regiszterpárban kell eltárolni. Ha a két adatblokk a memóriában nem fedi át egymást, úgy semmi problémánk nincs, amennyiben átfedés van a blokkok között, úgy alapvető feltétel, hogy a 'DE' által meghatározott célhely címe alacsonyabb legyen, mint az átmozgatandó blokk báziscíme, ellenkező esetben 'utolérjük magunkat', és ez beláthatatlan következményekkel járhat.

Az utasítás:

ED B0	237,176	LDIR	;blokk-mozgatás címnöveléssel

Mi történik valójában az LDIR utasítás kiadásakor? Az utasítás végrehajtásakor a 'HL' regiszterpár által meghatározott memóriahely tartalma a 'DE' regiszterpárban meghatározott memóriahelyre kerül, majd a 'BC' regiszterpár értéke eggyel csökken, a 'HL' és 'DE' regiszterpár értéke pedig eggyel nő. Ez a folyamat egészen addig tart, amíg a 'BC' regiszterpár értéke el nem éri a zérust. A végállapotban a 'HL' és a 'DE' regiszterpár értéke a blokkok utáni memóriahelyre mutat.

Tekintsük meg az utasítás használatát egyszerű példák bemutatásával. A blokk-mozgató rutint helyezzük el a 60000. memóriacímtől kezdődően:

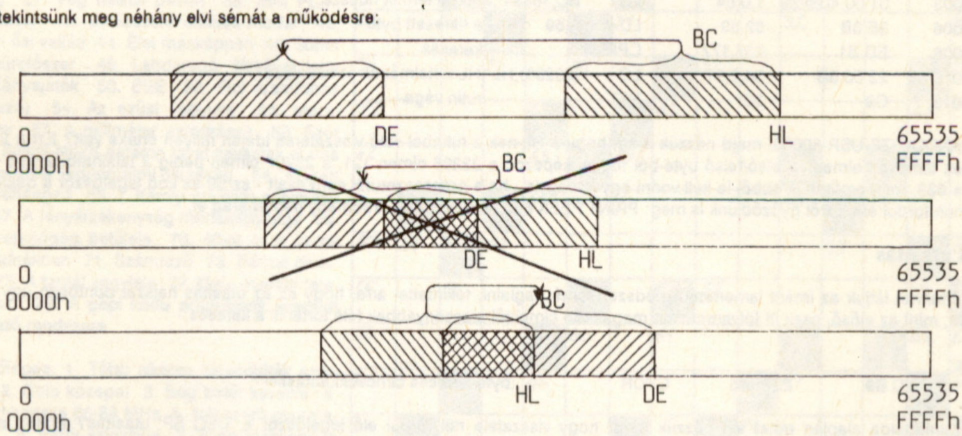
60000	21 00 40	33,0,64	LD	HL,16384	;felső 1/3 kezdőcíme
60003	11 00 48	17,00,72	LD	DE,18432	;középső 1/3 kezdőcíme
60006	01 00 08	1,0,8	LD	BC,2048	;blokk hossza
60009	ED B0	237,176	LDIR		;blokkmozgatás
60011	C9	201	RET		;rutin vége

A rutinból jól látható, hogy az átmozgatandó blokk startcímének a képernyő felső 1/3 kezdőcímét adtuk meg, a célhely címe pedig a középső képernyő harmad kezdőcíme lett. Tekintettel arra, hogy a blokk hossza éppen egy képernyő-harmadnyi, ezért gyakorlatilag a képernyő felső harmadát másoljuk át a középső harmadba, természetesen az eredeti állapot is megmarad. A példa szemléltetésére rajzoljunk egy kört a felső képernyő harmadba: CIRCLE 125,150,20, majd - ha a rutin már a memóriában van -, adjuk ki: RANDOMIZE USR 60000. A kör a középső harmadba átmásolódik. Az előzőleg említett átfedési problémát is tekintsük meg jelen példánk módosításával. Az átmozgatandó byte-ok számát változtassuk meg 2048-ról 4096-ra, vagyis ez azt jelenti, hogy a képernyő felső 2/3-át akarjuk átmásolni az alsó 2/3-ra. Változtassuk tehát meg a 60008. cím tartalmát 8-ról 16-ra: POKE 60008,16, majd rajzoljunk egy - a felső 2/3-ot átfedő kört a képernyőre: CIRCLE 125,120,50. Adjuk ki ismét: RANDOMIZE USR 60000. Nos hogy keletkezett 1 körből egy utasításra három körcikk? Egyszerű a magyarázat, amikor a byte-számláló a második harmad elejére ért, ott már az előbb átmozgatott adatokat találta, tehát ezt pakolta át a harmadik szeletbe. Ha a művelet fordítva történik, vagyis az alsó 2/3-ot helyezzük át a felső 2/3-ba, akkor a mozgatás hibamentes lesz. Cseréljük tehát fel a HL és DE regiszterpárok tartalmát: POKE 60002,72: POKE 60005,64,

rajzoljuk fel az előző kört az alsó 2/3 mezőbe: CIRCLE 125,60,50, majd RANDOMIZE USR 60000, és a kör 1/3 mezővel feljebb lépett. Természetesen az alsó 1/3 mezőben ott marad az információ, mert mint említettük az utasítás hatására az eredeti blokk adatai nem törlődnek, csak ha a blokk-mozgató művelet átfedéses és felülírjuk ezeket.
Több játékprogramban a screen betöltés közben hirtelen 'kiugrik' a képernyőre. Ott is ezt a rutint használják fel, a báziscímnek tetszőleges RAM címet, célcímnek a képernyő kezdőcímét, hossznak pedig egy screen hosszát (6912) beállítva. Gyakorlásképpen próbáljuk ki ezt, töltsünk be egy tetszőleges screen-t a memóriába, majd mozgassuk át a képernyő-memória területére.

b) LDDR utasítás

Ismét tekintsünk meg néhány elvi sémát a működésre:



Itt is látható, hogy az átmozgatandó blokk kezdőcímét a 'HL' regiszterpárban, a célhely címét a 'DE' regiszterpárban, míg az átmozgatandó byte-ok számát a 'BC' regiszterpárban kell eltárolni. Ha a két adatblokk a memóriában nem fedi át egymást, úgy itt sincs semmi probléma, úgy is mondhatjuk ez esetben teljesen tetszőleges, hogy melyik utasítást használjuk a kettő közül. Átfedés esetén alapvető feltétel, hogy a 'DE' által meghatározott célhely címe magasabb legyen, mint az átmozgatandó blokk báziscíme, ellenkező esetben megint csak 'utolérjük magunkat', a következményeket már ismerjük.

Az utasítás:

ED B8	237,184	LDDR	;blokk-mozgatás címcsökkentéssel

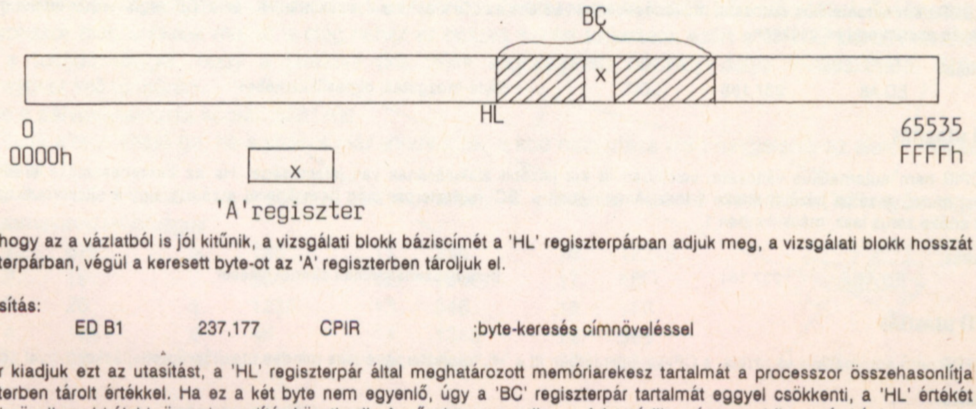
Mi történik, ha kiadjuk az LDDR utasítást? Végrehajtáskor a 'HL' regiszterpár által meghatározott memóriahely tartalma a 'DE' regiszterpárban meghatározott memóriahelyre kerül, majd a 'BC', 'HL' és 'DE' regiszterpárok értéke eggyel csökken. Ez a folyamat egészen addig tart, amíg a 'BC' regiszterpár értéke el nem éri a zérust. A végállapotban a 'HL' és a 'DE' regiszterpár értéke a blokkok előtti memóriahelyre mutat.

Itt gyakorlatilag az előző eset fordítottja fog történni. Írjuk be az előző mintapélda utolsó állapotába az LDIR utasítás helyére az LDDR utasítást, vagyis a 176-os kódot változtassuk meg 184-re: POKE 60010,184, majd - 'visszaszámlálás' lévén itt a képernyő 2/3 mezők végcímét kell megadnunk: HL-be 22527-et, DE-be pedig 20479-et: POKE 60001,255: POKE 60002,87: POKE 60004,255: POKE 60005,79, ezt követően ismét CIRCLE 125,60,50, majd RANDOMIZE USR 60000, és ismét láthatjuk az előbb már megismert hibajelenséget, csak most fordítva. A hiba kijavítása a 'HL' és 'DE' regiszterpárok felcserélésével történhet.

Az itt megismert két utasítás **jól kiegészíti egymást** minden irányban és átfedéssel történő blokk-mozgatás esetén.

c) CPIR utasítás

Amíg az előbbi két utasítás egy meghatározott memóriablokk mozgatását végezte el, addig ez és a következő egy általunk keresett adat-byte első előfordulását keresi meg:



Mint ahogy az a vázlatból is jól kitűnik, a vizsgálati blokk báziscímét a 'HL' regiszterpárban adjuk meg, a vizsgálati blokk hosszát a 'BC' regiszterpárban, végül a keresett byte-ot az 'A' regiszterben tároljuk el.

Az utasítás:

ED B1	237,177	CPIR	;byte-keresés címnöveléssel

Amikor kiadjuk ezt az utasítást, a 'HL' regiszterpár által meghatározott memóriarekesz tartalmát a processzor összehasonlítja az 'A' regiszterben tárolt értékkel. Ha ez a két byte nem egyenlő, úgy a 'BC' regiszterpár tartalmát eggyel csökkenti, a 'HL' értékét pedig eggyel növeli, majd újabb összehasonlítás következik. A művelet automatikusan folytatódik egészen addig, amíg két byte nem egyezik,

vagy a 'BC' el nem éri a zérust. Az első esetben a zérus jelzőbit 1-be, a második esetben 0-ba vált, a 'HL' regiszterpár pedig a keresett byte utáni rekeszre mutat. Ezután már a 'HL'-t csak be kell olvasnunk egy szabad memóriahelyre, ahonnan egyszerűen ki tudjuk a keresett értéket olvasni. Az itt kialakult címnél eggyel kevesebb jelöli a számunkra értékes rekesz címét.

Nézzünk a kereső mechanizmusra egy egyszerűbb példát: keressük meg egy rutinnal, hogy a ROM-ban hol található az első 59-es (decimális) kódú utasítás. Tekintettel arra, hogy a 'DEC SP' utasítást elég ritkán szokták használni.

```
60000   21 00 00   33,0,0    LD    HL,0         ;startcím a ROM kezdőcíme
60003   01 00 40   1,0,64    LD    BC,16384     ;a ROM terület hossza
60006   3E 3B      62,59     LD    A,59         ;a keresett byte
60008   ED B1      237,177   CPIR               ;keresés
60010   22 00 5B   34,0,91   LD    (23296),HL   ;HL tartalmának átmásolása
60013   C9         201       RET                ;rutin vége
```

Adjuk ki RANDOMIZE USR 60000, majd nézzük meg, hogy a HL-nek a rutinból való visszatérés idején milyen értéke volt? Ezt a 23296 és 23297-es címeken elmentett alsó/felső byte-ból fogjuk képezni: a 23296 címen 121, a 23297 címen pedig 2 található, 2*256+121 eredménye 633, mint említettük ebből le kell vonni egyet, vagyis - ha a processzorunk nem csalt - az 59-es kód legelőször a decimális 632-es címen fordul elő. Erről győződjünk is meg: PRINT PEEK 632, meglátjuk az eredmény nem marad el.

d) CPDR utasítás

Nem sok értelmét látjuk az imént ismertetett módszert ismét taglalni, tekintettel arra, hogy ez az utasítás hajszál pontosan ugyanazt hajtja végre, mint az előző, csak itt folyamatosan magasabb címektől alacsonyabbak felé történik a keresés.

Az utasítás:

```
ED B9      237,185      CPDR      ;byte-keresés címcsökkentéssel
```

Az előző mintapélda alapján most azt nézzük meg, hogy visszafelé hol fordul elő legelőször a 'DEC SP' utasítás? Ehhez át kell állítanunk a báziscímet, vagyis a 'HL' tartalmát a ROM végére: POKE 60001,255: POKE 60002,63, a CPIR utasítást pedig le kell cserélnünk a CPDR-re: POKE 60009,185. Adjuk ki: RANDOMIZE USR 60000, majd PRINT PEEK 23296, PRINT PEEK 23297. A kapott értékek: 193/59-ből képzett cím 15297, csak most ehhez hozzá kell adnunk egyet, mivel az előző utasítás fordítottja történik. Ez azt jelenti, hogy hátulról leghamarabb a 15298-as ROM címen fordul elő a 'DEC SP' utasítás. Erről is győződjünk meg: PRINT PEEK 15298.

2. Nem automatikus blokk-kezelő utasítások

Az ide tartozó négy utasítás működésében megfelel az előzőeknek, de itt egyidejűleg csak egy byte kezelés hajtódik végre, ezt követően a programozó döntheti el, hogy újra végrehajtja az utasítást, vagy nem. Ebből adódóan a nem automatikus blokk-kezelő utasítások végrehajtási ideje állandó.

a) LDI utasítás

Ez az LDIR nem automatikus változata, végrehajtásakor egy byte adat áttöltése történik a 'HL' regiszterpárban meghatározott rekeszből a 'DE' regiszterpárban meghatározott memóriarekeszbe, miközben a 'BC' regiszterpár értéke eggyel csökken, a 'HL' és 'DE' regiszterpárok értéke eggyel nő, valamint a túlcsordulás/paritás jelzőbit értéke mindaddig 1 lesz, amíg a 'BC' regiszterpár értéke zérusnál nagyobb. Ha ezt követően újabb LDI utasítást adunk ki, úgy a folyamat ismét végrehajtódik.

Az utasítás:

```
ED A0      237,160      LDI       ;byte-mozgatás, címnöveléssel
```

Általában az LDI utasítást követően egy vizsgálatot szokás elhelyezni a soron következő byte-ra vonatkozóan, pl. a zérus byte-okat figyelmen kívül hagyjuk, csak a többit mozgatjuk át, stb.
Ezeket az utasításokat már ritkábban használjuk gépi kódú programjainkban.

b) LDD utasítás

Ez az LDDR nem automatikus változata, működésében ekvivalens az LDI utasítással, csak itt a 'HL' és a 'DE' regiszterpár értéke minden végrehajtás esetén eggyel csökken.

Az utasítás:

```
ED A8      237,168      LDD       ;byte-mozgatás, címcsökkentéssel
```

c) CPI utasítás

Ez a CPIR nem automatikus változata, valójában itt két jelzőbit alakulásának van jelentősége. Ha az összehasonlítás eredménye egyezést mutat, a zérus jelzőbit értéke 1 lesz. Amennyiben a 'BC' regiszterpár még nem érte el a zérust, úgy a túlcsordulás/paritás jelzőbit értéke zérus lesz, máskülönben 1.

Az utasítás:

```
ED A1      237,161      CPI       ;byte-összehasonlítás címnöveléssel
```

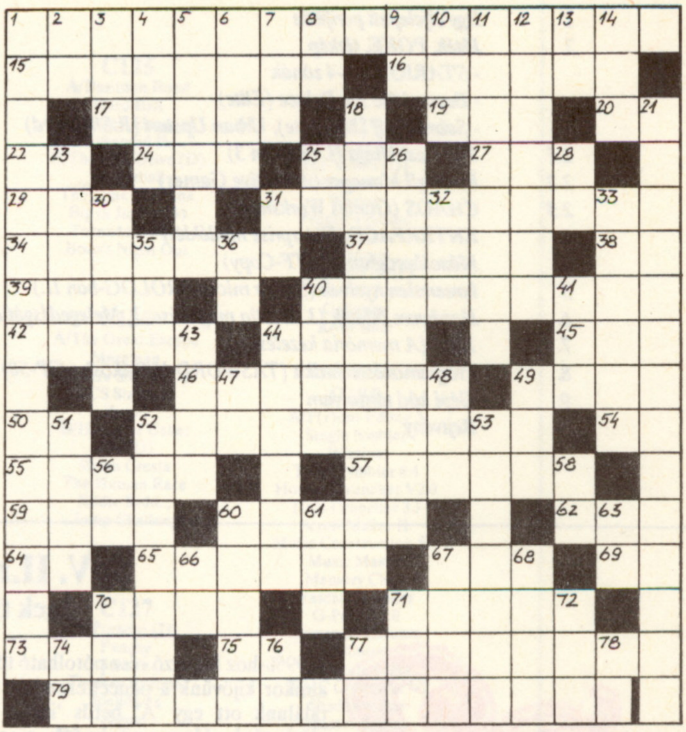
d) CPD utasítás

Ez a CPDR nem automatikus változata, a CPI-vel ellentétben itt a 'HL' regiszterpár értéke minden utasításvégrehajtáskor 1-gyel nő.

Az utasítás:

```
ED A9      237,169      CPD       ;byte-összehasonlítás címcsökkentéssel
```

PÁLYÁZATI REJTVÉNY



Vízszintes: 1. **Az ipari robotok végrehajtó-beavatkozó szerve** 15. Olasz napilap 16. Az ENSZ kulturális szervezete 17. Ritka f.finév 19. Járom 20. Az oxigén és a wolfram vegyjele 22. Orion hangfal típus betűjele 24. Számjegy 25. Fülöp szigetek-i és belga autók nemzetközi jelzése 27. Az ittrium és a neptunium vegyjele 29. Névelők 31. Kikísérletezi 34. Csehszlovák főzőedény 37. Vég nélküli trezor! 38. IBM PC/... 39. Huzal 40. Verébszerű énekes madár 42. Japán ősi vallás 44. Élet másképpen 45. Super ..... súrolószer 46. Labdarugó, társával tette 49. Kártyajáték 50. EEE 52. Pici áramforrás (két szó) 54. Az ezüst vegyjele 55. Mezei növény 57. A miliméter ezredrésze 59. Szór 60. A Svájci-NSZK határon tavat neveztek el róla 62. Lengyel légiforgalmi társaság 64. Kiejtett mássalhangzó 65. I. Károly idejéből származó jog 67. A fényérzékenység mértékegysége 69. SZU repülőgép betűjele 70. 40-et ér a dargli kártyajátékban 71. Származó 73. Párizsi divatcég 75. A tantál vegyjele 77. Győr melletti településen 79. **A gépi kódú programok elterjedt fejlesztő-rendszere**

Függőleges: 1. **Több sikeres kalandjáték szerzője** 2. Cola közepe! 3. Bog betűi keverve 4. Hullámhossza 66-88 MHz 5. Népszerű gyurmafigura 6. Fagy szlovákul 7. Anna másképpen 8. Bíztató szócska 9. Pályaudvar röv. 10. Indiai hírügynökség 11. Hűvös légárammal frissíteni 12. Nyugat-bengáliai város 13. Kamerun nemzetközi gk. jelzése 14. Azonos betűk 18. **Jól kivitelezett sportszimuláció** 21. **Magasszintű programozási rendszer** 23. Se ..., se száma! 25. Ludolf féle szám 26. ..... Leonard, (1918-) amerikai zeneszerző 28. Azonos a függ. 25-tel 30. Leánynév névelővel 31. Kicsinyítőképző 32. Igekötő 33. Város Washington államban 35. Nem itt! 36. Építőanyag 40. Utazás angolul 41. Mindig kevés van belőle 43. Fűtőanyag 44. Csavar 47. Ő németül 48. Alkalmas röv. 49. R.M.O. 51. Ábel testvére 52. 120 másodperc 53. Jobb erőben van 56. Reno egynemű betűi 58. Holland autók nemzetközi jelzése 60. Sár szlovákul 61. Hatalmas testű kutya 63. Lola egynemű betűi 66. Rangjelzés 67. Mars betűi keverve 68. Női név 70. Óriáskígyó 71. E.I.E. 72. Svéd váltópénz 74. Lám 76. Belga folyó 77. A kálium és a kén vegyjele 78. Kiejtett mássalhangzó

Beküldendő a **vízszintes 1., 79.**, valamint a **függőleges 1., 18. és 21.** sorok. A helyes megfejtést beküldők közül 5 nyertesnek elküldünk egy-egy darab - általuk előre kiválasztott, és a megfejtéssel együtt megjelölt - SPECTRUM vagy COMMODORE **programkazettát!**

Megfejtéseket csak a SpV következő részének megjelenéséig fogadunk el.

A nyereményeket postázzuk!

Manic Miner

A hagyományos (létrás) játékok kedvelőinek szeretnénk kedvezni az itt közölt információkkal. Tekintettel arra, hogy sokszor még örökélet mellett is elég nehéz az összes szintet teljesíteni, ezért a MANIC MINER rendelkezik azzal a tulajdonsággal, hogy a játékból (kis trükk segítségével) bármely pálya kiválasztható. Ehhez a következőket kell tennünk:

Írjuk be a pályakiválasztás kódját: 6031769

A kód csak akkor írható be, ha a játékot már elindítottuk, a kód megadása után megjelenik az életek mellett egy bakancs is.

Ezután be kell állítanunk a megfelelő pályát, amely a pályához tartozó kód-szám billentyűk egyidejű megnyomásával lehetséges:

1.	678	6.	136	11.	246	16.	12346
2.	16	7.	236	12.	1246	17.	56
3.	26	8.	1236	13.	346	18.	156
4.	126	9.	46	14.	1346	19.	569
5.	36	10.	146	15.	2346	20.	1256

Megjegyezzük, hogy a MANIC MINER mindkét verziójára hatásos az itt közölt módszer.

Tartalomjegyzék

SpV. II.rész, 6.oldal
Jack the Nipper



A 99%-hoz hiányzó rész pótolható többféleképpen is: - a bilit összetörhetjük; - amikor kijövünk a páncélteremből, és a szekrény tetején találjuk magunkat, találunk ott egy 'A' betűs 'izé'-t. Ez felvehető, és ha leejtjük 1-2 %-ot elérhetünk. Ha nem akarjuk magunkat agyonfárasztani, egyszerűbben is eljárhatunk: miután az Óvoda II.-ben felduáltuk a macskát a plafonra, dudáljunk tovább, ami természetesen nem hat kedvezően a jelenlévők egészségi állapotára, mivel a NAUGHTYOMETER rohamosan növekszik. Ezzel a módszerrel bármekkora értéket beállíthatunk a műszeren, így kevesebb teendőnk lesz a játék végigviteléhez.

SpV. 11.rész, 18.oldal
ENTERPRISE BASIC programok

A 3. programhoz (triangle demo) gépeljük hozzá a következő sorokat:

```
221 LET G=RND(50)
222 LET H=RND(50)
240 LET X4,Y4=140
370 PLOT #CHAN+2:X1,Y1;X2,Y2;X3,Y3;X4,Y4;X2,Y2;X4,Y4;X1,Y1
430 LET X1=X1+A:LET Y1=Y1+B:LET X2=X2+C:LET Y2=Y2+D:LET X3=X3+E:
    LETY3=Y3+F:LET X4=X4+G:LET Y4=Y4+H
491 IF X4>1279 OR X4<0 THEN LET G=-G:LET X4=X4+G
492 IF Y4>719 OR Y4<0 THEN LET H=-H:LET Y4=Y4
```

A módosítást elvégezve térbeli háromszöget mozgathatunk a képernyőn.

SpV. 13.rész, 5.oldal
DIZZY

A szárazjéggel nem a barlangban lévő 'csermelyt' kell befagyasztani, hanem az (L,3)-mas pályán lévő vízesést. A vízesés mögött egy labirintus van, amelyben a leghajmeresztőbb ötletek tömkelegével találkozhatunk. Ha sikerül végigvergődni a labirintuson, megszerezhetjük a csákányt, amivel az (F,6)-os pályán lévő omladékot távolíthatjuk el. Az omladék mögött lévő ékköves tőrrel (jewelled dagger) a (C,4) pályán lévő tutajt indíthatjuk el. A varázsló halálával gyakorlatilag még nem ér véget a játék, a patkót kell használni az asztal bal oldalán...

A címlapterv: Kodreán Zoltán munkája
A Spectrum Világ eddig megjelent részei utánvéttel megrendelhetők a következő címen:
Spectrum Világ, 1519 Budapest, Postafiók 363.

Felelős kiadó: Rucz Lajos, Székely László
K. F. eng.-sz.: 16 820
10 000 - 907306 Pátria Nyomda
Felelős vezető: Vass Sándor vezérigazgató

C121
A/5th Gear
Hideous Bill
Doctor Who
Wimbledon 64
B/Bazooka Bill
The Last V8
Fist II.
Flashdance (D)
Thrust Concert (D)

C122
A/City Cobra
Rambo
Break Dance
Gnom
B/Stop the Express
Inter.Karate I.
The Human Race
Kettle Game

C123
A/Inter.Tennis
Frak!
Outlaw's Flight
Paratroopers
Encounter
Bumping Buggies
B/Track & Field
Mooebius
Master Wrestling
Gandalf

C124
A/The Force
Wild Ride
Run for Gold
Formula I. Simulator
B/Ring Master
Inheritance
Elektra Glide
Jolly Roger

C125
A/Basildon Bond
Hard Ball
Desert Hawk
Arkanoid II.
B/The 1541 Drive (D)
Basketball
The Time Machine
Big in Japan (D)
Tornado Attack
Bozo's Night Out

C126
A/The Great Escape
Mazeman
Future Shock (D)
GCS Slamball
Zyron
B/House of Usher
Eye (D)
Moon Cresta
The Human Race
Eddie Kidd...
...Jump Challenge

C127
A/Porsche (D)
Punchy
Stealth
Theatre Europe
Spelldiver
Tapper
B/Little Pacer (D)
Agent U.S.A.
He Man
Fist Tour
Lazy Tunes (D)

C128
A/Axel F. (D)
Rudi Rat
BC. Bill
Squash
Bounty Bob
B/Space Pilot II.
Infiltrator II.
Black Hawk
Macadam Bumper

C129
A/Fighter Pilot
The Model (D)
B.McGuigan Boxing
Warlok
Frantic Freddie
B/Le Mans
Highnoon
Knucker Hole
Seawolf Joy
Gangster
River Raid

C130
A/F(r)ont Editor 3
Single Nibbler
Wizzine
Turbo Nibbler 4.4
Hotline Cruncher V2.0
Time Cruncher 3.1
Scroll Maker II.
Music Construction Set
Music Master
Memory Chart
Pascal Compiler
G-Pascal 3.0
Screen 80 Column
Sidmon
Magazine Index
B/MC Converter V2
Shadowwriter
Master Editor
Manager
Address Book
Superkit
Superkit II.
Speech V1.0
Power Windows
Speedscript
dBase de Luxe
Record Albums
Home Secretary
Text 64
Library Cards

A 'SpV' 16. részében megjelent keresztrejtvény helyes megfejtései:
Vízsz.16.: BASIC, 17.: FORTRAN, 39.: SIMSCRIPT, 72.: COMPILER, **Függ.**7.: ALGOL, 42.: LISP, 59.: COBOL
A 15. rész szerencsés nyertesei: B.Á. - Keszthely (S75), B.L. - Százhalombatta (S70), K.J. - Pécs (S92), S.A. - Budapest XI. (S111), S.K. - Budapest X. (S92); **nyereményüket postáztuk!**

TISZTELT OLVASÓ!

Köszönjük a bizalmat, hogy minden - a létező összes játékprogrammal kapcsolatos - kérdéssel felkeresnek bennünket. A szellemi és időbeni kapacitásunk azonban sajnos véges, így irreálisnak tűnik, hogy mindazokat a problémákat orvosolni tudjuk, amelyek az egyes játékok menetében önöket írásra késztetik. Önök is ismerik, milyen nehéz feladat egy-egy játék menetének a 100 %-ig történő kifejtése, így amit megismerünk, közkinccsé is tesszük a 'SpV' hasábjain. Kérjük Önöket, hogy a továbbiakban ilyen jellegű kérdésekkel **ne terheljenek** bennünket! **Köszönjük megértésüket!**